软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007621] [CSTR: 32375.14.jos.007621]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

基于异构依赖网络的开源 AI 资源可信性风险量化*

姚思梦[1,2], 张 洋[1,2], 赵佳林[1,2], 李俊辰[1,2], 沈 阳[1,2], 王 涛[1,2], 张迅晖[1,2]

[1](国防科技大学 计算机学院, 湖南 长沙 410073)

[2](复杂关键软件环境全国重点实验室, 湖南 长沙 410073)

通信作者: 张洋, E-mail: yangzhang15@nudt.edu.cn

摘 要: 随着开源模型与数据集规模快速扩张, Hugging Face 生态中资源之间形成了复杂的模型-数据集为主体的异构依赖网络, 元数据缺失、依赖集中度等问题使链式风险更易累积与传播. 为刻画这一风险基础, 基于 Hugging Face 快照构建了一个 AI 资源依赖网络, 并从全局拓扑与时间演化两个维度分析其结构和演化特征; 进一步提出融合属性完整性、社区反馈的“可信风险”指标, 对模型与数据集节点进行连续化风险量化与排序. 结果表明, 该依赖网络呈显著“尖峰+长尾”结构, 依赖高度集中于少数枢纽节点, 大量资源在关键数据流关系中处于孤立或半孤立状态; 同时, 模型规模爆炸式增长而数据集与贡献者扩张滞后, 导致生态对有限核心数据源的路径依赖持续增强, 形成结构性系统风险. 在节点层面, 可信风险指标对参数扰动保持稳健, 能够在多类风险来源下显著区分高/低风险节点并优于基线方法; 风险耦合分析与专家盲评进一步验证了高风险数据集与高风险模型在局部结构中的聚集与传播效应. 为开源 AI 生态的风险筛查与治理提供了可复现的量化依据.

关键词: 资源依赖网络; 依赖风险评估; 模型生态系统; Hugging Face; 异构依赖网络

中图法分类号: TP311

中文引用格式: 姚思梦, 张洋, 赵佳林, 李俊辰, 沈阳, 王涛, 张迅晖. 基于异构依赖网络的开源AI资源可信性风险量化. 软件学报. http://www.jos.org.cn/1000-9825/7621.htm

英文引用格式: Yao SM, Zhang Y, Zhao JL, Li JC, Shen Y, Wang T, Zhang XH. Quantifying Credibility Risk of Open-source AI Resources Based on Heterogeneous Dependency Networks. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7621.htm

Quantifying Credibility Risk of Open-source AI Resources Based on Heterogeneous Dependency Networks

YAO Si-Meng[1,2], ZHANG Yang[1,2], ZHAO Jia-Lin[1,2], LI Jun-Chen[1,2], SHEN Yang[1,2], WANG Tao[1,2], ZHANG Xun-Hui[1,2]

[1](College of Computer Science and Technology, National University of Defense Technology, Changsha 410073, China)

[2](State Key Laboratory of Complex & Critical Software Environment, Changsha 410073, China)

Abstract: As the scale of open-source models and datasets continues to expand, the Hugging Face ecosystem forms a complex heterogeneous dependency network centered on model-dataset relationships. Issues such as missing metadata and high dependency concentration make chain-structured risks more likely to accumulate and propagate. To characterize this underlying risk landscape, this study constructs a resource dependency network of open-source AI resources based on a Hugging Face snapshot and analyzes its structural and evolutionary characteristics from the perspectives of global topology and temporal evolution. Furthermore, a “credibility risk” indicator is proposed by integrating metadata completeness and community feedback, enabling continuous risk quantification and ranking of model and dataset nodes. The results show that the dependency network exhibits a pronounced “spike-and-long-tail” structure, in which dependencies are highly concentrated on a small number of hub nodes, while a large proportion of resources remain isolated or semi-isolated within critical data-flow relationships. Meanwhile, the explosive growth of models, combined with the relatively slow expansion of datasets and contributors, strengthens the ecosystem’s path dependence on a limited set of core data sources, thereby giving rise to

* 基金项目: 国家自然科学基金青年基金 (62202480)
收稿时间: 2025-09-17; 修改时间: 2025-11-09; 采用时间: 2025-12-17; jos 在线出版时间: 2026-04-22

structural systemic risks. At the node level, the proposed credibility risk indicator demonstrates robustness to parameter perturbations and effectively distinguishes high-risk from low-risk nodes across multiple risk sources, outperforming baseline methods. Risk-coupling analysis and expert blind evaluation further confirm the clustering and propagation effects of high-risk datasets and high-risk models within local structures. Overall, this study provides a reproducible quantitative basis for risk screening and governance in open-source AI ecosystems.

Key words: resource dependency network; dependency risk assessment; model ecosystem; Hugging Face (HF); heterogeneous dependency network

开源模型生态近年来进入高速扩张期, 平台化托管推动模型与数据集的沉淀与复用, 由此在资源层面形成规模化、结构化的依赖网络[1]. 以 Hugging Face (HF) 为例, 截至 2025 年 3 月已累计托管超 150 万个模型与 33 万个数据集. 在这样的生态中, 单个资源 (模型或数据集) 的健康状况会沿依赖链条传播, 影响可复现性、合规性与社区信任, 并抬升平台治理与审查成本[1−3]. 因此, 一个迫切的问题是: 模型与数据集分别面临哪些“资源级问题”, 这些问题在依赖网络中如何分布与传播?

与传统软件供应链[4−14]中基于明确版本与依赖声明 (如“包-版本-导入路径”) 的分析框架不同, 开源模型生态中的依赖关系呈现出显著的跨类型特征. 此类依赖不仅包括模型之间的微调与继承, 还广泛涉及模型对训练数据集的引用、数据集之间的衍生与扩展等复杂链路. 然而, 这些依赖信息大多散布于模型卡 (model card)、数据集卡 (dataset card) 的非结构化文本或标签字段中, 缺乏统一、可解析的版本化声明机制. 这一根本差异使得传统基于“显式依赖图+版本约束”的软件供应链分析方法难以直接适用, 也导致整个模型生态在依赖可见性与风险可溯性方面存在严重缺失.

在此背景下, 构建能够系统刻画模型、数据集、作者、任务与标注者等多类实体间语义关系的异构依赖网络, 成为理解生态依赖结构、识别潜在风险传播路径的基础. 才能从全局视角分析资源之间的归属关系、任务分布与来源依赖, 进而评估由关键节点或链路所引入的集中性风险与传播脆弱性.

在厘清网络拓扑结构的基础上, 节点级别的可信状态评估成为风险识别的关键补充. 拓扑位置仅能反映节点在依赖传播中的潜在影响范围, 而节点的真实风险还深受其属性完整性 (如许可证、作者信息)、社区反馈强度 (如下游依赖、点赞) 等因素影响. 因此, 需要发展能够融合多源信息的量化评估方法, 以支持对关键风险节点的精准识别与预警.

尽管关于模型生态的安全、可持续性、文档质量与社区协作等议题已有初步探索[15−19], 但仍缺少基于真实元数据、面向生态整体的依赖网络建模与面向资源可信复用的量化工具, 致使平台与研究者在规模审查的流程中, 缺乏低成本、可解释的前置筛查信号.

为此, 本文基于 HF 公开元数据构建模型-数据集异构依赖网络刻画网络结构特征与潜在脆弱点, 并提出面向大规模初筛的可信风险指标, 该指标专注于资源许可证与来源字段完整性、社区反馈以及其在依赖网络中的结构位置等数据, 与传统的安全/合规/性能风险相关但不同, 其定位为在大规模生态中用于排序与预警的前置信号评分.

我们的分析显示, HF 生态的模型-数据集依赖网络呈现典型的“尖峰+长尾”结构,依赖高度集中于少数枢纽节点且大量节点在关键数据流关系中处于孤立状态.与此同时,模型数量的爆炸式增长远超数据集与维护者扩张速度,使生态对少数核心资源的路径依赖不断强化,从而积累显著的系统性结构风险. 指标评估结果进一步表明,该可信风险指标稳健且具有良好判别力,能够有效识别关键风险节点.

本研究的主要贡献如下.

(1) 提出基于真实元数据的生态级依赖网络建模方法, 并据此给出模型与数据集的脆弱性洞察.

(2) 构建了一种具有可解释性的“可信风险”评估指标, 综合考虑属性完整性、社区反馈等多个关键因素, 用于全面衡量生态节点的可信度与依赖风险. 此外, 本文已开源所构建的数据集与分析工具, 相关资源链接: https://www.gitlink.org.cn/YovM/AIResourceRiskTools.

本文第 1 节介绍本研究的背景与相关工作. 第 2 节给出研究方法. 第 3 节基于所构建的网络, 刻画 AI 资源 (模型、数据集) 依赖网络的基本特征,并给出依赖带来的可信性风险量化结果. 第 4 节讨论本研究的实践启示. 第 5 节对全文进行总结.

1 背景与相关工作

本文从两个关键视角对相关研究进行系统回顾: (1) 开源生态系统中的依赖关系; (2) 依赖关系的风险度量与评估方法.

1.1 开源生态系统中的依赖关系

已有研究基于 PyPI、NPM、Cargo 等生态构建显式依赖图, 主要聚焦于具备版本控制特性的代码依赖, 常见研究内容包括传递性依赖深度以及依赖关系的集中度等信息[11,12,20,21]. 这些研究揭示了依赖网络结构的演化过程对系统性风险的影响, 一些工作还进一步探讨了维护者与平台的互动关系[17,18], 但局限于代码级依赖.

近年来, 开源 AI 生态的依赖更为复杂, 开源 AI 生态的依赖对象不仅是模型-模型的派生/复用, 还包含模型-数据集的使用关系, 以及数据集-数据集的继承/派生关系, 且大量依赖以非结构化文本出现. 已有尝试包括基于 HF 元数据的供应链图谱, 但主要关注结构特征[22]以及许可现状[23]等局部问题; 而 Yang 等人[24]则针对特定子域 (面向代码任务的大模型) 进行手工整理依赖与许可合规分析. 总体来看, 现有研究多围绕依赖网络中的单一侧面展开, 缺乏统一的风险导向评分框架. 与之不同, 本文在同一异构图中联合建模模型、数据集与相关元信息, 为后续风险度量提供结构基础.

1.2 依赖关系的风险度量与评估方法

在软件包生态中, 研究以中心性/路径深度/传递性等指标来表征“一处失效, 处处受影响”的传播潜势; 不少工作用 CVE/OSV 数据量化直接与间接漏洞对下游的影响, 并把依赖网络指标用于缺陷预测[25], 以及分析关键依赖包失效所引发的系统性脆弱性[11]. DevSecOps 和医疗健康系统中, 故障模拟被广泛应用于捕捉关键组件失效对系统整体安全性与可靠性的影响[26,27]. 这类方法说明: 结构位置是“前置风险信号”的重要来源, 可用于排序与预警, 另一些工作从维护行为与生存性刻画“项目健康度”, 如用“可变更性”“可复用性”以及“潜在影响力”等指标, 用以量化软件包的更新频率、复用程度[11]. 另有研究引入生存分析方法, 考察软件包或项目活跃度的时间演化过程, 并进一步分析组织支持对项目可持续性的影响[12]. 但上述对象主要是代码包, 且特征与标签体系与模型生态并不等同.

针对数据/模型文档, 社区提出 datasheets for datasets[28]与 model cards 框架[29], 并有大规模实证揭示字段缺失与实践差异[30,31]; METRIC 等框架尝试量化数据质量[32]; 工程溯源工具 (如 MLflow2PROV 等[33]) 进一步捕捉数据-模型-元数据的生成过程. 它们表明可追溯性和文档完整度是可信复用的重要前置信号, 但未与网络结构位置结合, 亦缺乏面向生态级风险排序的量化方法.

与上述工作不同, 本文将属性完整性、社区反馈同依赖网络的结构特征结合统一到可信性风险指标中, 用于生态治理中的前置筛查与优先级排序.

2 研究方法

本节首先阐明研究动机与研究问题, 随后介绍研究的数据采集与预处理流程、异构依赖网络的构建方法、RQ1 所需的结构与演化特征分析方法、RQ2 的可信风险指标设计及其有效性实验配置, 并进一步说明节点风险分析与可视化工具的实现. 整体研究流程如图 1 所示.

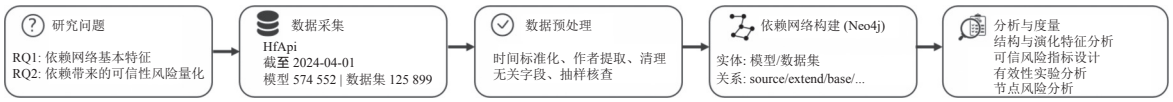


图 1 本研究的整体流程框架

2.1 研究问题

随着 HF 等开源平台的快速扩张, 模型与数据集不再是孤立资源, 而是通过训练来源、继承微调与数据衍生等机制形成跨类型的依赖网络. 由于该网络缺乏统一的版本追踪与依赖治理, 关键节点一旦出现合规或质量问题, 可能沿依赖链产生放大式传播. 因此我们确立如下问题.

RQ1: AI 资源依赖网络具有什么基本特征?

现有开源模型研究多聚焦于单资源属性或下载/点赞等外显指标, 但对“模型-数据集跨仓库依赖”所形成的整体结构缺乏系统刻画. 在没有全网拓扑与演化证据的情况下, 难以判断生态是否存在枢纽过度集中、长链传播或维护失衡等系统性脆弱点. 因此我们首先提出 RQ1, 旨在从网络规模与组件结构、节点度分布、依赖路径深度、时间演化等维度揭示 HF 依赖网络的结构与演化特征.

RQ2: 如何量化 AI 资源依赖带来的可信性风险?

RQ1 的拓扑结果无法直接刻画单节点的真实风险, 因为节点本身还深受其属性完整性 (如许可证、作者信息)、社区反馈强度 (如下游依赖、点赞) 等因素影响, 为在大规模生态中识别关键风险节点, 我们提出 RQ2 构建指标, 并对其稳健性与判别力进行系统验证.

2.2 数据采集

本研究所使用的数据来源于 HF 平台. 该平台汇集了丰富且更新频繁的开放数据资源, 具备高度的多样性与代表性[34]. 因此, 本文选取 HF 平台作为研究对象, 以全面刻画开源模型生态系统中模型与数据集之间的依赖关系及其演化特征.

我们首先参考了 HFCommunity 数据集[35], 该数据集覆盖了 HF 平台上的 3 类资源库: 模型、数据集与空间应用, 并包含标签、讨论、提交记录、作者信息等详细数据. 然而, 该数据集存在一些局限性, 例如不包含资源库的创建时间戳, 限制了对平台增长过程的深入分析, 且数据更新截至 2023 年 10 月.

为了确保数据的时效性和全面性, 本研究并未选择 HFCommunity 数据集, 而是通过 HF 官方提供的 HfApi 接口 (数据接口链接: https://huggingface.co/docs/huggingface_hub/main/en/package_reference/hf_api#api-dataclasses) 重新抓取了数据. 具体来说, 我们以平均每天 10 万条记录的速度进行数据抓取, 截至 2024 年 4 月 1 日, 共采集到 574 552 个模型条目和 125 899 个数据集条目.

在数据预处理阶段, 我们将 createdAt 字段转换为标准时间格式, 以便进行时间序列分析. 同时, 我们从 HF 提供的模型卡和数据集卡中提取了作者信息, 并删除了稀疏或无关字段 (如 SHA 哈希值以及不相关标签, 如“tag-TikTok”“tag-Twitter”等).

2.3 依赖网络构建

为构建 AI 资源依赖网络, 本文选用了 Neo4j[36]图数据库作为底层数据管理平台. 该平台擅长处理复杂关系数据, 能够有效支持模型生态系统中依赖结构的表达与建模. 图 2 展示了本研究构建的资源依赖网络的整体架构. 依赖网络在构建完成后的整体规模与各类节点分布情况将在第 3.1 节中详细报告, 作为后续结构特征分析的基础.

在本研究中, 模型与数据集是本研究关注的核心对象, 我们主要围绕其依赖关系与使用过程中可能存在的风险进行分析, 包括数据来源的可追溯性、依赖路径的冗余性以及许可证信息的完整性等. 作者与标注者则作为辅助信息引入, 旨在补充相关背景信息, 如作者身份是否明确等, 以便从更全面的角度理解依赖管理中的潜在风险.

上述实体属性主要来源于 HF 平台上的标签信息, 这些标签提取自模型卡与数据集卡. 相关信息构成了资源依赖网络的基础, 有助于刻画模型生态系统中的数据流动与依赖结构. 统计数据显示, 提供模型卡与数据集卡信息的模型与数据集占 HF 平台总下载量的 90% 以上[31,37]. 这一事实表明, 本文所使用的标签信息具有较高的覆盖率, 所构建的资源依赖网络在代表性与有效性方面均具有可靠保障. 同时, 本研究已采取措施应对 HF 元数据的潜在不精确性. 首先, 我们随机抽取了 100 个节点进行手动核查, 确保依赖关系的准确性, 结果未发现明显的错误或虚假关联.

本研究在构建资源依赖网络时, 定义了以下节点和关系.

(1) 节点类型定义

网络包含以下节点类型.

• Model: HF 上的模型仓库.

• Dataset: HF 上的数据集仓库.

• Author: 模型/数据集的作者或组织维护者.

• Annotator: 数据集标注/创建来源.

• Task: 任务领域节点, 用于任务聚合与跨任务比较.

• Placeholder: 用以代表一个在平台快照中缺失或不可追踪的资源实体, 从而保证指向该资源的依赖关系得以保留.

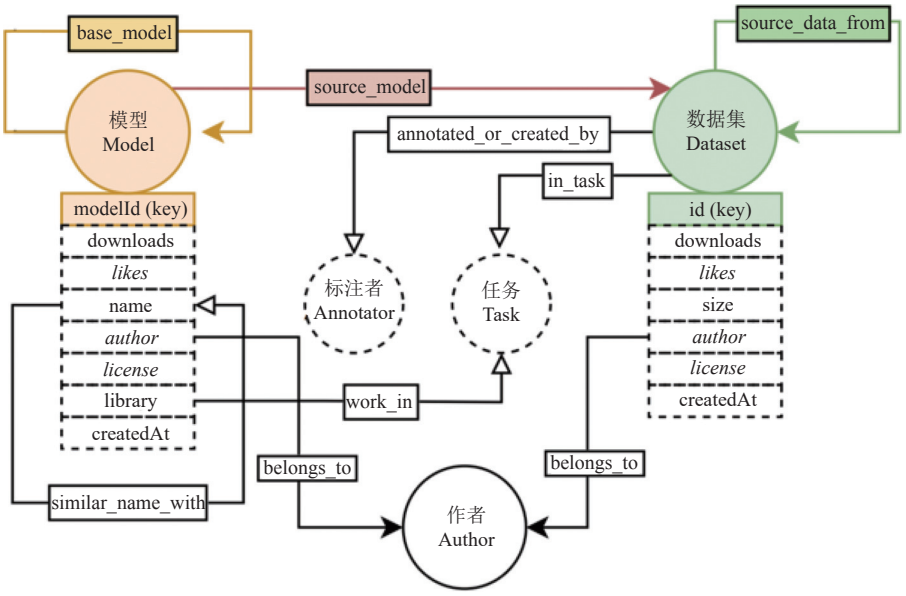


图 2 依赖网络总体框架

(2) 关系类型与语义

网络边类型与方向定义如下.

• belongs_to (Model/Dataset→Author): 资源归属作者/组织.

• annotated_or_created_by (Dataset→Annotator): 数据集标注/创建来源.

• in_task (Model/Dataset→Task): 资源所属任务领域.

• work_in (Annotator→Task): 标注/创建来源参与的任务领域.

• source_model (Model→Dataset): 模型训练/构建依赖的数据集来源.

• source_data_from (Dataset→Dataset): 数据集直接来源于其他数据集.

• extend_data_from (Dataset→Dataset): 数据集扩展/派生自上游数据集.

• base_model (Model→Model): 模型继承/微调自上游基础模型.

在本研究中, 异构依赖网络的构建遵循自底向上的统一流程. 首先, 我们以 HF 快照清单作为基准节点集合, 为所有模型与数据集创建唯一实体节点, 同时对作者、任务、标注者等附属实体进行初始化. 随后, 对资源卡片与配置文件中的显式依赖线索进行解析: 从模型侧提取 tag-dataset 与相关字段实例化 source_model 关系, 由继承信息构建 base_model 依赖; 从数据集侧解析数据来源与扩展字段, 分别实例化 source_data_from 与 extend_data_from; 并依据模型的 pipeline_tag 与数据集的 tag-task_categories 映射资源到所属任务, 建立 in_task 关系; 同时根据数据集的 tag-annotations_creators 生成 annotated_or_created_by 边, 并通过该关系与任务信息联立推导标注来源的任务覆盖, 形成 work_in 关系.

为避免依赖链在解析过程中因目标资源缺失而被中断, 我们进一步引入占位节点机制: 当某条依赖边指向的资源在快照中不存在, 或由于重命名而无法对齐为平台实体时, 为其生成 Placeholder 节点并保留原始依赖指向,

以最大程度保留依赖结构的连续性与潜在传播路径.

通过上述流程, 我们最终得到一个覆盖多实体类型、具有丰富语义关系的 HF 异构依赖网络. 此外, 为捕捉依赖关系在不同时间段的动态演化过程, 我们基于实体的创建时间戳将网络划分为多个时间窗口, 从而观察模型生态中资源依赖结构随时间的变化趋势.

2.4 结构与演化特征分析

RQ1 关注“资源依赖网络具有什么基本特征”. 为系统刻画开源模型生态的依赖形态与潜在脆弱性, 本研究从“结构特征”与“演化特征”两个互补维度展开分析.

在静态拓扑层面, 结构性风险强调依赖网络由于拓扑不均衡而可能产生的系统性不稳定, 包括依赖是否过度集中于少数枢纽节点、传播路径是否存在过长的链式结构, 以及是否存在大规模稀疏或断层区域等. 结合复杂网络理论的经典结论[38], 我们采用层次化视角对结构性风险进行度量.

其一, 通过统计全量网络中不同节点/关系类型的规模、连通组件的数量与大小分布, 并记录占位节点的比例, 来评估生态依赖的全局覆盖范围及潜在碎片化程度. 其中, 占位节点仅在关键依赖解析时生成, 其增量定义为“构图后节点规模-原始快照规模”, 分别在模型侧与数据集侧计算, 以估计关键依赖链中不可追踪资源的下界. 同时, 我们进一步在任务与贡献者粒度上统计模型/数据集的任务分布 (模型侧依据 pipeline_tag, 数据集侧依据 tag-task_categories) 以及标注来源的任务覆盖范围 (由 annotated_or_created_by 与 in_task 联立推导 work_in), 以揭示不同领域的依赖参与程度与贡献集中度差异.

其二, 在局部重要性层面, 我们围绕关键依赖关系 (extend_data_from、source_data_from、source_model、base_model) 对子图中的模型与数据集节点开展入度/出度统计与分布分析, 以刻画依赖复用是否过度集中于少数枢纽节点. 具体而言, 本文分别计算模型侧与数据集侧在关键依赖子图上的度值分布, 绘制直方图并统计低度节点占比与高度节点贡献的连接比例, 用以识别“尖峰+长尾”的连接异质性形态. 同时, 在尾部区间对度分布进行幂律拟合并报告幂律指数[39]以形式化刻画其长尾程度. 基于度中心性排序, 进一步提取连接度排名靠前的枢纽节点并汇总其度值, 用于揭示潜在的“单点依赖”风险. 在跨类型依赖方面, 构建模型-数据集二部子图, 仅保留模型侧显式声明的数据集来源关系 (source_model). 在该二部图上, 分别统计“每个模型直接依赖的数据集数量”与“每个数据集被模型依赖的数量”的分布特征与均值水平, 并绘制对数尺度直方图, 用以刻画跨类型显式依赖的整体分布形态与复用结构.

其三, 在传播通道层面, 多跳依赖路径的深度与数量用于刻画风险传播通道的长度及其潜在放大空间, 考虑到依赖边具有方向性且网络整体非强连通, 我们仅沿依赖方向对关键依赖边子图进行有界遍历 (bounded BFS), 最大跳数设为 10, 并只统计简单路径的数量. 路径统计分别在两类关键关系链上独立进行: 模型侧的 base_model 继承链, 以及数据集侧的 source_data_from 与 extend_data_from 组成的 source(extend)_data_from 衍生链. 我们在 1–10 跳范围内逐跳记录可达路径数量及其随跳数的增长/衰减趋势, 并提取最长可观测依赖链长度, 用以衡量模型与数据集依赖在多跳传播上的深度差异与影响半径. 通过该多跳路径计数, 我们能够从“链条结构”尺度补充揭示依赖网络的潜在传播通道与结构性风险.

通过上述 3 层指标, 我们能够在全局规模、节点集中度与多跳链条这 3 个尺度上对静态拓扑下的结构性风险形成一个相对完整而可操作的刻画.

在时间维度层面, 演化性风险关注生态扩张过程中“资源规模变化-维护主体变化-依赖关系变化”之间的动态演化过程. 具体地, 我们以月为粒度跟踪关键实体与关系的动态变化: 一方面, 统计模型、数据集与作者等节点规模的月度增长曲线, 以对比不同主体的扩张速度; 另一方面, 统计核心依赖边 (如 source_model、base_model、source(extend)_data_from) 的月度增量及其相对占比变化, 用以刻画依赖模式的时间演化趋势; 同时比较新增资源与新增作者之间的增速差异, 评估维护主体增长与资源扩张之间的动态关系.

2.5 可信风险指标设计

仅有拓扑结构统计仍不足以衡量单节点的实际可信性风险: 合规元数据缺失与社区验证不足等微观因素会与

拓扑位置共同决定其风险暴露程度. 因此, RQ2 进一步引入了“属性完整性” (特别是许可证与作者信息) 以及“社区反馈”等关键维度, 构建了一个综合性的量化指标——“可信风险”. 该指标旨在从微观层面评估节点在其外部环境与依赖网络中的整体可信度, 并提供初始的参考风险值, 以辅助研究者与维护者在实际应用中开展更有针对性的筛查与预警工作.

(1) 风险维度

• 属性完整性. 在对模型与数据集的整体属性进行调查时, 我们发现元数据缺失问题不仅限于版本记录、任务类型或描述性字段. 对于开源平台而言, 许可证信息与作者信息通常是与合规性最直接相关的核心要素[19,40]. 一旦许可证信息缺失或失效, 模型或数据集的复用与分发将面临较大的不确定性, 可能引发侵权风险, 并增加后续维护与更新的复杂度. 同时, 缺乏明确的作者声明也使责任归属难以追溯, 妨碍平台对侵权或合规纠纷的及时响应与处理.

• 社区反馈. 在开源生态系统中, 点赞数量与下游依赖数量等指标可作为衡量节点使用热度与外部审查程度的重要参考. 当一个节点同时具有较高的点赞数和大量下游依赖时, 通常意味着其获得了更广泛的社区认可, 并具备较强的持续维护支持能力[41]. 生命周期校正的反馈强度. 不同节点的存在时间会影响累计 *likes* 的可比性: 新资源因时间较短往往偏低, 早期资源则可能因累积效应偏高. 为消除这一时间偏置, 我们使用 $likes/\Delta t$ 对社区反馈信号进行生命周期归一化, 以刻画单位时间内的平均外部验证强度.

(2) 形式化定义

因此, 本文提出一种多因素度量框架, 以辅助识别潜在的高风险节点. 在实际实现过程中, 为使指标对重大合规性缺陷具备更高敏感性, 同时能够平滑地反映累积性或渐进式风险, 我们采用指数函数、对数函数与乘除式函数相结合的方式进行度量.

首先, 许可证与作者信息是开源资源最关键的合规性元数据, 其缺失通常构成“灾难敏感型”风险: 一旦出现, 将直接影响资源的再分发、商用许可以及责任追溯. 与一般风险不同, 此类风险具有明显的“门槛效应”: 即使节点在其他方面表现良好, 只要存在关键元数据缺失, 其总体风险就应显著提升.

这一逻辑与金融风险管理中的光谱风险度量 (spectral risk measure, SRM)[42]高度一致. SRM 通常采用由指数效用函数诱导的权重函数, 对高损失尾部赋予急剧上升的权重, 用以体现对尾部不利事件的高度敏感性[42,43]. 借鉴这一建模思路, 本文在可信风险指标中采用指数函数对关键合规性缺失进行放大. 该结构可视为对“关键信息缺失”的指数惩罚与 SRM 中针对尾部灾难事件的处理方式在建模逻辑上保持一致.

大量用户行为研究指出, 流行度类指标 (例如内容热度、点赞数、浏览量等) 通常呈现高度偏斜的长尾分布结构, 即少数项目获得大量互动, 而大多数项目仅获得极少反馈[44,45]. Wu 等人[46]基于百万级用户数据的实证分析进一步揭示, 此类行为变量往往由“随机乘法增长”与“新奇性衰减”共同驱动, 因此整体上呈 log-normal 分布特征. 在这类分布中, 规模增长具有显著的递减边际效应, 即在小规模区域变化剧烈, 而在大规模区域变化趋缓.

基于这一成熟结论, 统计学习和信息系统研究普遍采用对数变换来处理行为型长尾变量, 以实现: ① 方差稳定化; ② 削弱极端值影响; ③ 更好地刻画规模效应的非线性; ④ 使变量在模型中的变化趋势更可解释.

在本研究的模型生态系统中, $likes/\Delta t$、out-degree 等变量均属于由用户行为驱动的量化反馈指标. 我们对其分布特性进行描绘, 观察到与典型用户行为数据相似的长尾结构. 因此, 在风险指标设计中, 我们遵循统计学中处理长尾行为变量的通用做法, 对这些量采用对数变换, 用以捕捉规模增长的递减边际效应并增强指标的稳健性.

在依赖拓扑维度上, 我们进一步假设下游依赖数量 *num* 可近似衡量节点的外部审查强度: 被更多模型所复用的节点在真实使用中更容易被持续验证, 因此其风险应随 out-degree 增加而下降. 为使该下降过程既单调又平滑, 本文在构造风险因子时将 out-degree 引入分母, 并加入 $1/(num+1)$ 的平滑项, 以避免 *num* 较低时出现数值突变, 使风险随 out-degree 的增长呈现连续、稳定且边际递减的下降趋势.

对数变换不仅用于弱化长尾分布中极端值的影响, 还具有经典的方差稳定化特性, 可使多数量级差异的行为特征在统一尺度上进行聚合建模, 从而符合一般风险模型对稳定输入分布的要求.

乘除运算则用于平衡不同风险维度之间的“累积效应”与“相互抵消”关系, 使高风险因素得到放大, 同时使诸如下游广泛验证等正向因素有助于降低整体风险. 据此, 我们构建了节点可信风险的度量函数, 如公式 (1) 所示.

$$F^{a}_{\mathrm{cred}}(\mathrm{author},\mathrm{license},\Delta t,\mathrm{likes}_{num},num)=\mathrm{e}^{\alpha[S_a(\mathrm{author})+S_a(\mathrm{license})+\ldots]}\times\frac{1}{\ln\left[\mathrm{e}\times\beta\times\left(\frac{\mathrm{likes}_{num}}{\Delta t}+1\right)\right]}\times\frac{1}{\ln\left[\frac{\mathrm{e}}{2}\times\left(\frac{1}{num+\gamma}+num+\gamma\right)\right]} \quad (1)$$

其中, $S_a(\cdot)$ 为布尔函数, 用于判断对应属性是否缺失, 取值为 0 或 1. 分母部分的函数用于衡量节点所受到的外部审查程度; 当当前节点没有任何入边 (即无下游节点指向该节点) 时, 该函数取值为 1.

许可证缺失 $S_a(license)$: 当该值为 1 时, 表示节点缺失关键信息, 如许可证或作者信息, 其风险将通过指数函数进行放大.

Δt: 表示节点从创建/最近更新到快照时, 节点的生命周期长度.

$likes_{num}$: 表示节点在社区中获得的总点赞数.

num: 表示依赖该节点的下游节点数量. 较大的 num 值通常意味着更严格的外部监督与验证, 从而在分母中降低整体风险值.

(3) 高/低风险阈值

为实现模型与数据集在全网范围内具有可比性的风险分层, 本文依据各自的全局风险得分分布设定统一的分位数阈值. 具体而言, 我们对模型节点与数据集节点分别计算其全体风险值的分布, 并将全局 80% 分位数作为高风险界, 将全局 20% 分位数作为低风险界; 得分不低于高分位阈值的节点被标记为高风险, 不高于低分位阈值的节点被标记为低风险.

2.6 有效性实验设置

为系统评估可信风险指标在大规模异构依赖网络中的可靠性与判别能力, 本文从参数稳健性、结构化正负样本、基线方法对比与专家评审这 4 个角度设计验证实验, 形成互补的有效性证据链. 整体目标是检验该指标是否能够在不同风险来源情境下稳定区分高风险与低风险节点, 并在连续排序层面提供超越单一属性的风险刻画能力.

(1) 参数稳健性

在参数稳健性层面, 为检验本文可信风险指标在不同参数设定下的稳健性, 我们进一步对公式 (1) 中 3 个关键调节项进行了系统的敏感性分析: ① 指数惩罚项的放大系数 α, 用于强调许可证与作者信息缺失所带来的“门槛型”风险; ② 对行为变量进行平滑处理的对数项系数 β; ③ 用于 out-degree 平滑的系数 γ (对应于 num+1 结构). 在实验中, 本文以 α=β=γ=1 作为基准设定, 并对上述参数分别施加±20% 的扰动, 比较扰动前后节点风险排序的一致性, 以检验可信风险指标对参数变化的稳健性.

排序一致性采用 Spearman 秩相关系数[47]与风险最高与最低的前 10% 节点集合的重合率作为评价指标, 从而判断排名结构在不同参数设定下是否保持稳定, 以验证可信风险指标的敏感性与稳健性.

(2) 结构化正负样本

为验证可信风险指标在不同风险来源下的敏感性与区分能力, 本文基于若干关键风险特征构建结构化的正负样本集合. 上述风险特征包括: 社区反馈强弱 (如点赞量高低、出度大小)、元数据完整性 (如许可证信息是否缺失、是否具有明确的基础模型标签) 等, 这些维度均被认为与开源资源的合规性、可维护性及复用风险密切相关.

在样本构建过程中, 针对连续型特征 (如 $likes$、out-degree), 本文分别从其分布两端抽取 30 个正样本与 30 个负样本, 以刻画不同风险水平对指标输出的影响; 而对于离散型特征 (如 $license$), 则依据是否具备完整许可证信息直接构建二元分组. 所有抽样均采用固定随机种子, 以确保实验过程的可复现性. 最终, 共构建覆盖 7 个风险维度的正负样本集合, 系统涵盖了模型与数据集在多类风险情境下的表现.

在统计检验方法上, 本文采用 Mann-Whitney U[48]非参数检验比较正负样本组的风险得分差异, 并报告 Cliff's delta (δ)[49]作为效应量, 用以评估差异的方向与强度. 显著性判断上, 当 p<0.05 时认为两组差异显著, p<0.01 与 p<0.001 分别表示更强与极强的显著性证据[50]. 非参数检验避免了正态性与方差齐性假设, 适用于 $likes$、out-degree 等长尾分布的风险特征; 效应量 δ 则用于刻画正负组风险得分在分布层面的可分性, 为指标判别能力提供量化依据. 依据效应量的常用解释标准[49], Cliff's delta 可分为 4 个范畴: 可忽略效应 ($|\delta|$<0.15)、小效应 (0.15≤$|\delta|$<0.33)、中等效应 (0.33≤$|\delta|$<0.47) 以及大效应 ($|\delta|$≥0.47). 效应量越大, 表示正负样本在风险得分上的差异越具可

分性, 说明指标的判别能力越强.

(3) 基线方法对比

为验证多因素建模的增益并剥离不同维度的贡献, 本文设置两类对比基线. 其一为 license-only baseline, 仅依据许可证/合规元数据是否缺失进行二值判定; 其二为 feedback-only baseline, 仅保留社区反馈相关连续项 (*likes*/Δt 与 out-degree), 用于检验单一外显行为信号是否能够替代完整指标. 基线方法与完整指标将在 Top-20 排名一致性及正负组区分能力等维度上进行对比, 以评估多因素融合建模带来的有效性增益.

(4) 专家评审

为了进一步验证风险指标在实际情境中的可解释性与应用价值, 本文在统计检验与基线对比的基础上, 补充开展了结构化的专家盲评实验. 该实验旨在评估指标在真实决策环境中的可理解性、排序合理性及其对风险来源的揭示能力.

具体而言, 我们依据风险得分分布进行分层抽样, 选取 10 个高风险节点与 10 个低风险节点构建评估集合, 并邀请 5 名从事模型开发与管理, 且具有 1 年以上开发经验的专家在盲评条件下独立打分. 所有节点均以标准化的“节点摘要卡”形式呈现, 包含统一的数据字段及说明, 评分采用 5 点 Likert 量表, 从“强烈不同意”至“强烈同意”, 涵盖 4 个维度: 节点被判定为“高/低风险”的合理性、指标对风险来源的可解释性、指标在筛查中的实用性以及当前排序位置的可接受度.

2.7 节点风险分析

在定义可信风险核心指标的基础上, 为全面评估其揭示的生态风险模式, 我们进一步从以下 3 个维度展开节点级别的风险分析.

(1) 风险耦合分析

风险耦合分析旨在探究高风险节点在依赖网络中是否倾向于相互连接, 从而形成局部的风险聚集区. 具体而言, 我们将所有模型-数据集依赖边 (source_model) 根据其两端节点的风险类型, 划分为“高-高”“高-低”“低-高”“低-低”这 4 个类别, 并构建 2×2 列联表. 通过卡方独立性检验[51]判断模型与数据集的风险状态是否关联. 同时, 计算每个数据集的风险得分与其所有下游模型平均风险得分之间的 Spearman 等级相关系数, 以量化风险在依赖链上的传递与耦合强度.

(2) 任务维度的风险分析

为探究不同任务领域是否呈现出差异化的风险暴露模式, 我们将模型与数据集映射到其对应的任务节点, 并在任务粒度上开展组间比较. 一方面, 计算各任务下资源的平均可信风险, 以衡量任务生态在整体风险水平上的差异. 另一方面, 进一步统计各任务中高风险节点的比例, 用以识别风险是在整体水平上抬升, 还是主要集中在尾部节点, 以揭示不同任务领域在风险分布形态上的潜在差异.

(3) 典型高/低风险节点识别与案例研究

为提供直观的风险画像并验证指标的实际洞察力, 我们基于风险得分排序, 分别筛选出模型与数据集中风险最高与最低的典型节点 (如 Top-5). 通过对比分析其在元数据完整性、社区反馈等方面的特征, 归纳高/低风险节点的共性模式. 并选取具有代表性的典型案例进行深入剖析, 以定性方式展示综合风险模式的识别过程.

2.8 可视化工具设计

在可视化工具的设计中, 本文预设了 3 类查询模式, 分别为: 关系聚焦查询、作者聚焦查询和节点路径查询. 关系聚焦查询用于探索特定类型关系在图结构中的分布与连接模式; 作者聚焦查询则以某一作者节点为中心, 展示其关联的模型与数据集构成的关系网络; 节点路径查询用于可视化任意两个节点之间的最短路径, 帮助理解实体之间的潜在依赖链条.

在图可视化生成方面, 本工具以 Neo4j 图数据库为后端基础, 前端渲染部分采用 Neo4j 官方提供的可视化库 Neovis.js (工具链接: https://github.com/neo4j-contrib/neovis.js/), 实现图结构的动态展示与交互集成. 目前, 该工具已开源, 支持本地部署 (项目地址为: https://www.gitlink.org.cn/YovM/AIResourceRiskTools).

3 研究结果

3.1 RQ1: AI 资源依赖网络具有什么基本特征?

(1) 网络规模与组件结构

表 1 与表 2 给出了依赖网络中各类节点与关系的规模统计. 最终网络包含 575 531 个模型节点、131 225 个数据集节点、172 729 个作者节点、90 个标注者节点及 112 个任务节点, 共计约 80 多万实体节点; 关系类型以 belongs_to、source_model 与 base_model 为主, 占据网络的主要连接骨架.

表 1 依赖网络中各类节点规模统计

节点类型	数量
Author	172 729
Annotator	90
Model	575 531
Dataset	131 225
Task	112

表 2 依赖网络中各类关系规模统计

关系类型	数量
belongs_to	700 451
annotated_or_created_by	4 998
in_task	24 905
extend_data_from	591
source_data_from	3 236
source_model	89 329
base_model	69 797

与原始快照相比, 模型节点规模仅增加 0.17%, 而数据集节点规模增加 4.23%. 由于新增节点全部由占位节点构成, 这一不对称的增幅说明: 在实际依赖建模过程中, 数据集相关依赖更频繁地指向“缺失或不可追踪的资源”, 例如外部数据源被删除、重命名后未同步更新, 或依赖对象仅在描述文本中被提及但未在平台上形成对应条目. 换言之, 相较于模型, 数据集在外部引用缺失、信息更新不同步或资源下线等问题上表现得更为突出, 从而在构图阶段体现为更显著的节点数量上升.

图 3 展示了从全量异构依赖网络中筛选得到的局部示意子图, 图中不同颜色表示不同节点类型: 蓝色为模型 (Model) 节点, 紫色为数据集 (Dataset) 节点, 两者分布最广, 是网络的核心主体; 绿色为作者 (Author) 节点; 橙色为任务 (Task) 节点, 用于连接同一功能类别下的模型与数据集、体现资源的任务聚合; 青绿色为标注者 (Annotator) 节点. 与此同时, 不同颜色的边对应表 3 中的关系类型: 绿色边为 in_task, 粉紫色边为 source_model, 蓝色边为 belongs_to, 橙色边为 annotated_or_created_by, 灰色边为 extend_data_from, 红色边为 base_model, 青绿色边为 source_data_from.

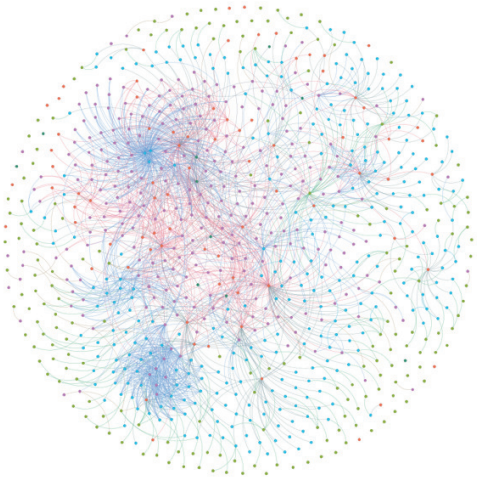


图 3 异构依赖网络子图概述

我们以高连接度/高活跃度的模型与数据集为种子节点, 并在关键依赖关系上进行有限跳扩展所得到的, 可以看到, 在这一局部子图中, 模型与数据集通过训练依赖、基础模型继承以及数据来源链路形成网络主干, 并围绕任

务节点聚合为多个局部簇团; 不同簇团之间又通过共享数据来源或基础模型形成跨簇连接, 这些依赖链路共同构成模型生态中的关键传播通道.

表 3　贡献者资源维护统计

作者	模型数量	数据集数量	资源总数	作者	模型数量	数据集数量	资源总数
CyberHarem	2939	3345	6284	liuyanchen1015	58	1558	1616
open-llm-leaderboard	0	5620	5620	Helsinki-NLP	1442	27	1469
TheBloke	3863	0	3863	stefan-it	1325	8	1333
huggingtweets	3830	0	3830	timm	1284	13	1297
tkcho	3809	2	3811	jonatasgrosman	1265	0	1265
Jeevesh8	3723	0	3723	cleanrl	1207	15	1222
autoevaluate	23	3232	3255	tyzhu	378	806	1184
sail-rvc	3182	0	3182	AdapterOcean	387	788	1175
LoneStriker	2895	0	2895	PrunaAI	1162	0	1162
LarryAIDraw	2661	0	2661	gokuls	1131	25	1156
Recag	8	2270	2278	hkivancoral	1155	0	1155
FounderOfHuggingface	2136	0	2136	testorga13h1	1089	0	1089
Facebook	1907	18	1925	YakovElm	1059	0	1059
KingKazma	1729	0	1729	sd-concepts-library	1029	4	1033

在作者维度上, 模型节点数量 (575 531) 与数据集节点数量 (131 225), 显著多于作者节点的数量 (172 729). 这一差异表明该平台高度依赖少数维护者资源支持, 平均每位维护者管理约 4.1 个资源.

进一步的统计结果显示, 排名前 1% 的维护者 (约 1 750 人) 共控制平台 34.2% 的资源, 体现出强烈的中心化特征. 其中, 排名第 1 的维护者 CyberHemer 单独管理了 6 284 个资源 (详见表 3). 表 3 罗列了资源数量超过 1 000 的核心作者, 进一步突出了模型生态中的集中化现象.

此外, 在标注来源维度上, 尽管 Annotator 节点在整体网络中占比较小, 但其对应的 annotated_or_created_by 关系极为密集. 仅有 90 个标注者节点就建立了 4 998 条标注关系, 以全网约 80 多万个实体节点规模估算, 这 90 个节点仅约占节点总数的 0.01%, 却承担了全部标注边. 这一高度集中的标注模式, 说明数据标注任务高度集中于少数贡献者之手. 此类集中化的标注行为可能导致标注偏差的出现. 当初始标注过程存在偏差时, 模型与数据集之间的协同演化机制可能会进一步放大并积累这些偏差, 最终在多轮训练中逐步加剧. 以计算机视觉领域的关键数据集 ImageNet 为例, 已有研究[52]指出其在某些类别的标注质量和代表性方面存在问题. 如果在关键类别的标注过程中存在偏差, 势必会影响基于该数据集训练的深度学习模型的性能与公平性, 甚至削弱其在实际应用中的有效性.

在任务粒度上, 各任务的资源规模与依赖形态存在显著差异. text-generation、text-classification 与 text2text-generation 等任务分别覆盖 71 309、54 125 与 25 868 个模型, 对应的数据集数量为 2 283、2 252 与 283. image-classification 与 text-to-image 等视觉相关任务也具有较大规模, 分别包含 10 930 与 18 823 个模型. 相比之下, reinforcement-learning 任务虽然包含 39 580 个模型, 但仅有 15 个数据集带有显式任务标注.

从关键依赖边的平均度来看, 不同任务之间也呈现结构差异. 如图 4 所示, automatic-speech-recognition (ASR) 任务的数据集平均度为 14.29, 是所有任务中最高的; image-classification (3.64)、fill-mask (3.92) 与 token-classification (3.98) 任务的数据集平均度也相对较高. 相比之下, text-classification (2.54) 与 text2text-generation (2.98) 等任务对应的数据集平均度相对较低.

进一步对比两端可见, 模型侧的任务间平均度差异相对较弱, 且整体连接水平偏低 (多数任务的模型平均度低于 1); 相较之下, 数据集侧的平均度在任务之间波动更明显.

综上, 各任务在资源规模、数据集平均度与模型平均度上均呈现可观察的统计差异. 数据集端的结构差异幅度显著大于模型端, 不同任务在依赖网络中的连接模式并不一致.

图 5 展示了 Annotator 在任务维度上的覆盖数 (work_in 出度) 的累积分布函数 (CDF). 从整体来看, 该分布呈

现出显著的集中性与长尾特征: 在全部 60 个能够与明确任务关联的 Annotator 中, 有 73.3% (44/60) 仅覆盖 1 个任务领域, CDF 在覆盖数为 1 时快速升至 0.73; 进一步地, 约 90% 的贡献者任务覆盖数不超过 2 个. 这说明大多数标注来源仅参与极少量任务, 其贡献具有明显的“单任务聚集”特征.

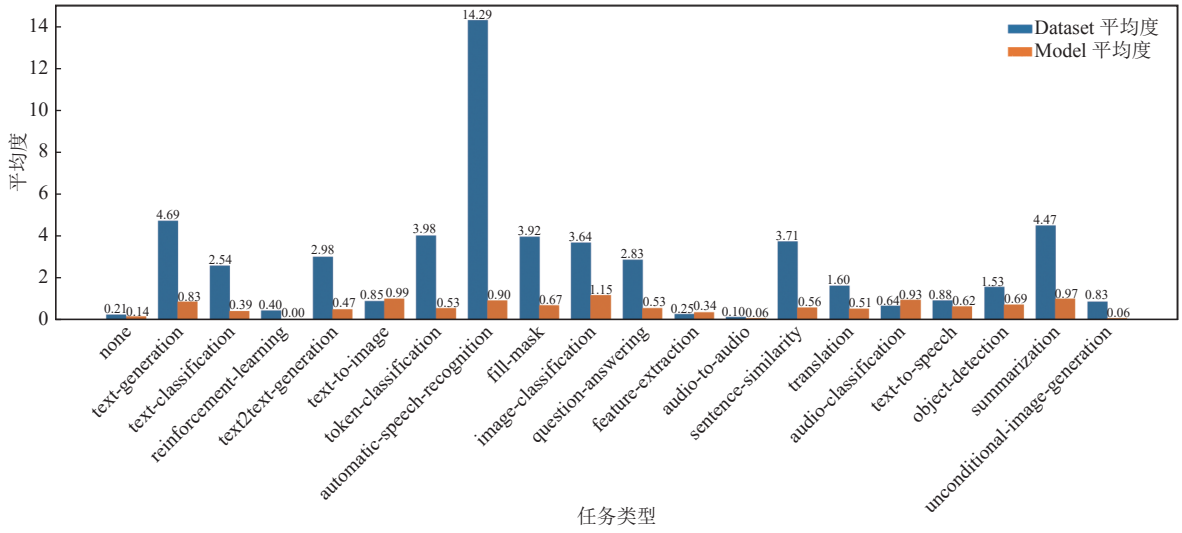


图 4　主要任务的数据集 (Dataset) 平均度与模型 (Model) 平均度对比结果

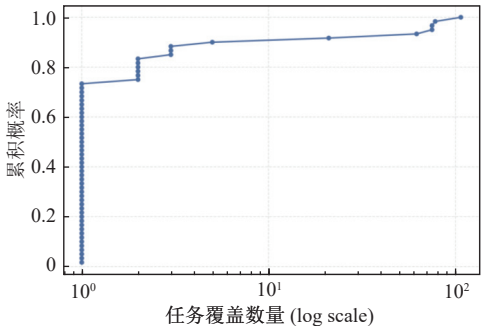


图 5　标注者任务覆盖数量分布的累积概率

与此同时, 分布右侧形成了由极少数节点构成的长尾区间, 个别跨任务贡献者的覆盖范围远高于平均水平, 最大任务覆盖数可达 107. 覆盖数的描述性统计也支撑这一点: 尽管中位数为 1、上四分位数为 2, 但均值上升至 8.13, 这表明长尾极端值显著拉高了整体平均水平. 进一步计算得到的 Gini 系数[53]为 0.798, 反映出任务覆盖在贡献者之间呈现高度的不均衡性.

为进一步分析长尾的构成, 图 6 展示了任务覆盖数量最高的 Top-20 Annotator. 可以看到, 任务覆盖数量最高的标注/创建来源主要集中在少数几类具有高度复用特征的方式, 例如 expert-generated、machine-generated、crowdsourced、found 等. 这些来源通常在多个任务中被重复使用, 因此在任务维度上形成较高的覆盖范围. 相比之下, 大多数 Annotator 的任务覆盖范围仅为 1–2 个任务, 呈现典型的“头部集中-尾部分散”结构.

综合从网络规模、任务分布与标注来源覆盖得到的结果可见, HF 模型-数据集依赖网络在宏观结构上呈现出明显的不均衡特征: 数据集侧的依赖链更容易指向不可追踪的外部来源, 导致占位节点集中出现; 资源的维护与标注贡献高度集中于少数核心主体; 同时, 依赖复用强度与标注覆盖在不同任务间表现出显著的头部集中与长尾异质性. 整体而言, 这些特征共同揭示了生态在全局覆盖、核心依赖关系与领域分布上的系统性不均衡结构.

发现 1: 数据集侧的依赖链更易指向不可追踪的外部来源, 少数核心维护者与标注者掌控了大部分资源与标

注关系, 致使生态依赖高度集中; 加之不同任务领域的依赖模式差异显著, 且标注活动呈现“领域内集中、领域间割裂”的格局, 共同加剧了生态的系统性风险.

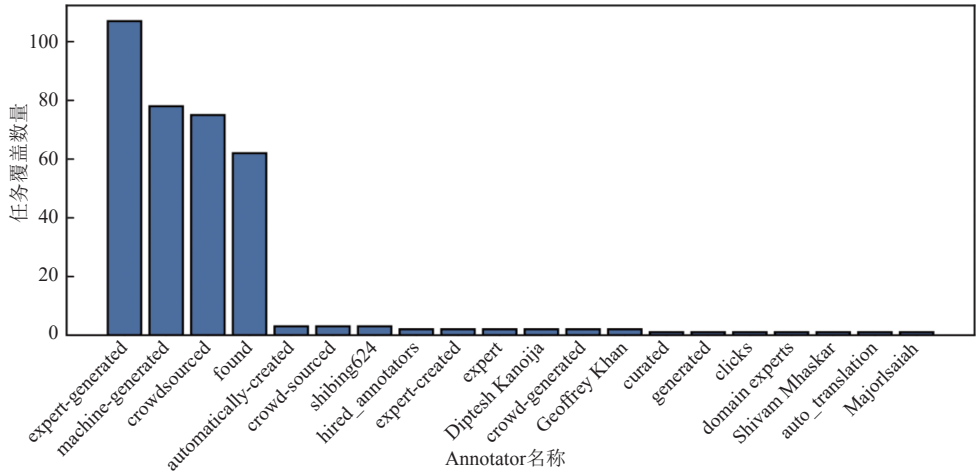


图 6　标注来源任务覆盖数量 Top-20 分布图

(2) 节点度分布

图 7 展示了模型与数据集在关键依赖关系上的入度/出度分布, 图 7(a) 采用对数坐标展示全范围度分布; 图 7(b) 采用线性坐标并聚焦低度区间 [1, 10) 用于占比对比.

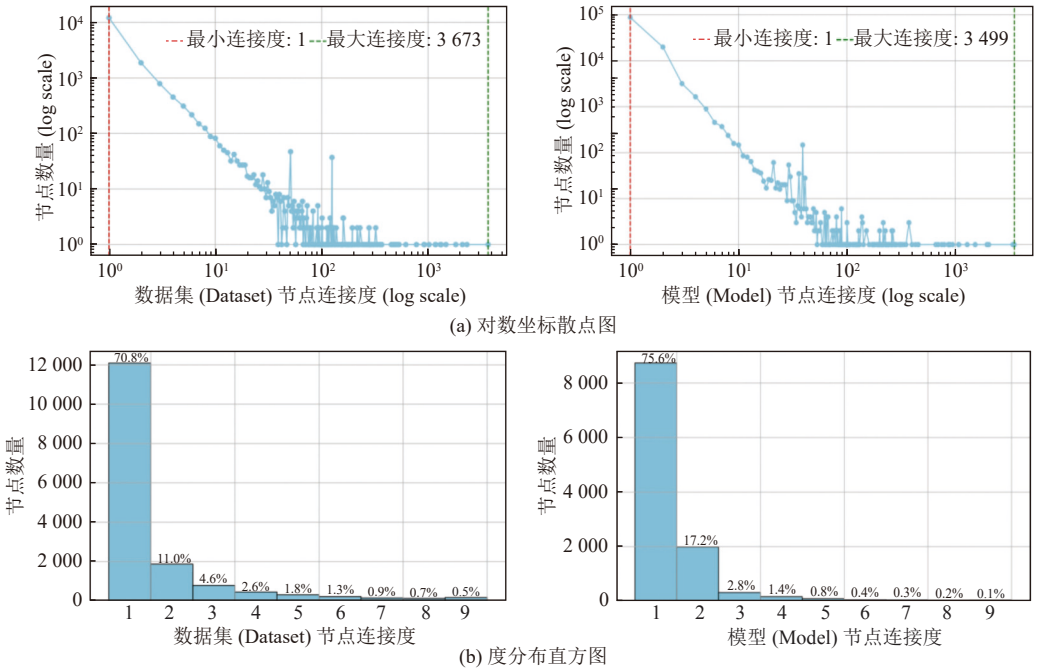


图 7　数据集和模型节点的连接度分布图

从图 7 中可以看出, 两类节点均呈现典型的“尖峰+长尾”结构, 度分布直方图揭示了网络连接的极度异质性: 在数据集节点中, 有 70.8% 的节点仅维持一条连接 (度=1), 而度大于 100 的高连接节点仅占 0.9%, 却贡献了全球 41.6% 的连接数. 其中, 核心枢纽节点 imagefolder 拥有高达 3 673 条连接. 类似的分布特征也出现在模型节点中:

75.6% 的模型节点度值仅为 1, 而仅有 0.09% 的高连接模型节点 (度>100) 却承担了网络中 16.6% 的总连接数. 其中, distilbert-base-uncased 是模型网络中的关键枢纽节点, 具有 3 499 条连接. 该类“尖峰+长尾”式的连接模式表明模型生态系统对少数关键节点存在显著的过度依赖.

经过对比发现, 数据集节点的网络中心性显著高于模型节点 (41.6% vs.16.6%). 这一差异可能源于权威数据集的高标注成本, 例如 ImageNet 包含约 1 400 万张人工标注图像[54], 而模型则可通过参数微调快速派生出大量变体, 使得核心数据集成为不可替代的关键节点. 这种结构性不平衡揭示出模型生态系统中严重的“单点依赖”风险, 一旦核心数据集发生异常, 可能引发大规模的级联式失效.

幂律拟合结果进一步验证了网络的长尾拓扑特征. 通过计算尾部累计分布函数 (CDF) 并结合幂律分布验证方法 (见公式 (2)), 我们观察到显著的长尾特性. 如图 7(a) 所示, 数据集节点与模型节点的度分布均呈现典型的幂律分布形态, 其幂律指数均接近于 1.17[39].

$$P(X) \propto X^{-\alpha} \tag{2}$$

分析表明, 尽管网络中不存在完全孤立的节点, 但在关键数据流关系 (包括 extend_data_from、source_data_from、source_model 以及 base_model) 中, 仍有大量节点处于孤立状态. 具体而言, 87.0% 的数据集节点与 80.0% 的模型节点未参与上述关键依赖关系. 这种在核心依赖关系中的广泛孤立现象, 导致整个网络在互联性与协同作用方面存在显著缺失. 综上所述, 模型生态系统中的主要风险来源于网络连接结构的不均衡, 以及对少数枢纽节点的过度依赖.

我们通过对节点的度中心性进行测量, 即识别连接数量最多的节点, 对结果进行了汇总, 并呈现在表 4 中. 鉴于数据规模庞大, 该表主要展示了连接数排名前 12 的模型与数据集节点. 我们观察到, 许多中心节点缺乏明确的作者信息, 甚至存在作者信息缺失的情况. 然而, 进一步核查后发现, 部分节点实际上具备明确的原始作者信息. 例如, “imagenet-1k”已被迁移并合并至“ILSVRC/imagenet-1k”, 尽管原始仓库已对相关信息进行了更新, 但依赖该仓库的模型或数据集未能及时同步其源信息, 导致作者信息缺失或链接失效.

表 4　连接数排名前 12 的模型与数据集节点度数

模型	节点度数	数据集	节点度数
distilbert-base-uncased	3 499	imagefolder	3 673
stabilityai/stable-diffusion-xl-base-1.0	3 391	glue	2 323
GPT2	2 033	original	2 106
runwayml/stable-diffusion-v1-5	1 958	squad	1 825
mistralai/Mistral-7B-v0.1	1 524	emotion	1 474
xlm-roberta-base	1 286	mozilla-foundation/common_voice_7_0	1 326
bert-base-cased	1 078	imdb	1 325
bert-base-uncased	1 068	imagenet-1k	1 307
openai/whisper-tiny	944	wikipedia	1 183
roberta-base	893	xtreme	1 170
openai/whisper-small	875	generator	1 024
meta-llama/Llama-2-7b-hf	870	mozilla-foundation/common_voice_11_0	924

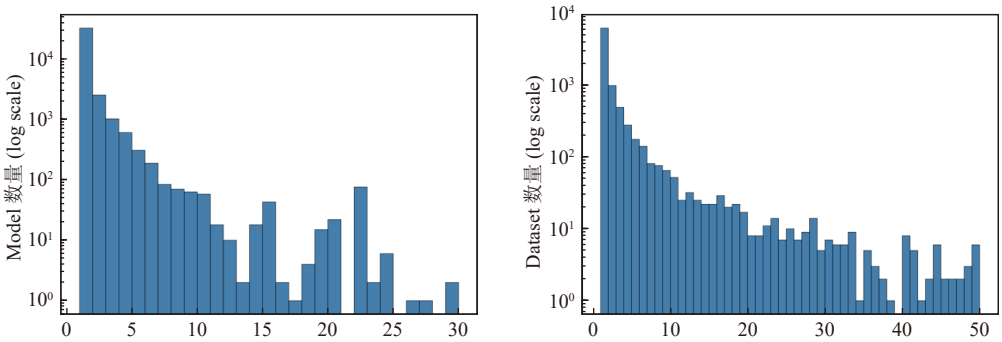
值得注意的是, 这些信息缺口并非均匀分布, 而是高度集中在连接度最高的枢纽节点上. 枢纽节点既承担着最密集的依赖关系, 也是元数据最容易积累缺失的位置, 使其成为网络中同时承载“信息风险”与“结构风险”的关键点.

跨类型依赖方面, 模型-数据集二部结构同样呈现复用集中. 我们在整体的模型节点与数据集节点共抽取到 60 351 条“模型→数据集”依赖边. 其中, 仅有 37 630 个模型与 9 200 个数据集至少参与了一条显式的模型-数据集依赖, 这说明当前生态中只有一部分资源在文档层面清晰标注了数据来源, 仍存在大量“隐式”或未记录的资源依赖信息.

从模型视角看, 每个模型直接依赖的数据集数量整体较少, 呈现显著的“单一依赖”特征: 在所有有依赖记录的模型中, 85.8% 仅关联 1 个数据集, 92.5% 的模型依赖不超过 2 个数据集, 平均度仅为 1.60. 不过仍有约 1.30% 的模型同时依赖 10 个及以上的数据集, 形成少量多源数据密集复用的长尾模型节点.

从数据集视角看, 则呈现出更强的“被复用集中”现象: 有依赖记录的数据集中, 虽然 67.6% 只被 1 个模型使用、78.3% 不超过 2 个模型使用, 整体看似较为分散, 但仍有约 7.43% 的数据集被 10 个及以上模型共同依赖, 平均每个数据集连接 6.56 个下游模型. 相关统计结果与计算脚本已随复现包一并提供, 便于审阅与复现.

图 8 展示了“每个模型依赖的数据集数量”和“每个数据集被模型依赖的数量”的对数坐标直方图: 整体趋势均在小度值处形成陡峭“尖峰”随后快速衰减, 并在高于 10 的区域拖出明显的长尾. 这一结果表明, 在模型-数据集二部图上同样存在典型的“尖峰+长尾”结构: 多数模型和数据集只参与局部、浅层的依赖关系, 而极少数高连接数据集承担了大量跨模型的共享复用.



(a) 每个模型依赖的数据集数量分布　(b) 每个数据集被模型依赖的数量发布

图 8　模型 (Model) 与数据集 (Dataset) 之间依赖关系的统计分布

综合同类型依赖与跨类型依赖的分析结果可以看出, 模型生态系统中的连接结构在多个层面均表现出显著的不均衡性和枢纽集中度. 少数高连接数据集和模型节点不仅在本类型网络中扮演关键角色, 在模型-数据集二部图中也形成明显的高复用核心, 从跨类型依赖维度进一步佐证了结构性风险集中于少数枢纽节点的结论.

发现 2: 模型与数据集均呈现强长尾与枢纽集中, 数据集侧集中度更高; 多数节点对核心依赖关系参与不足, 少数枢纽节点同时承载结构与信息双重风险; 此外, 跨类型依赖中亦呈现相同的长尾集中特征.

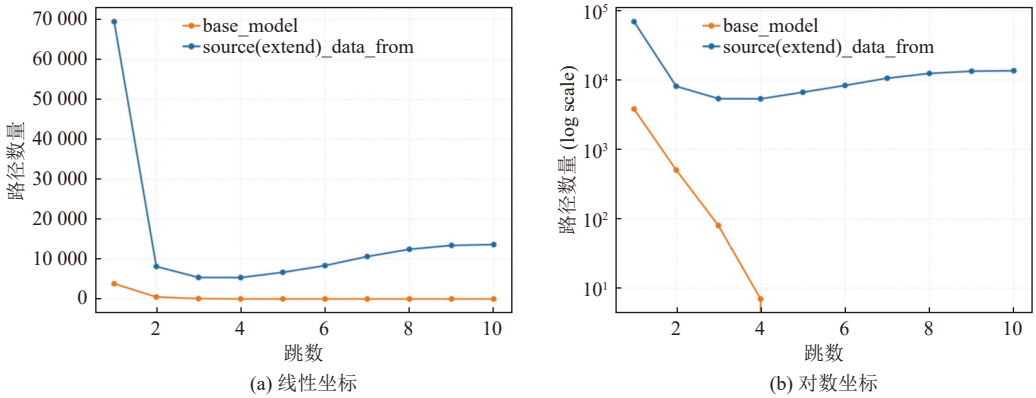
(3) 依赖路径深度

图 9 展示了模型继承链 (base_model) 与数据集衍生链 (source(extend)_data_from) 在多跳依赖下的路径数量变化. 对于 base_model 关系, 路径数量在 1–4 跳阶段呈现明显下降, 而在 5–10 跳阶段又出现显著回升, 整体表现为“先衰减、后反弹”的非单调趋势. 一方面, 1 跳路径数与边数近似相当, 说明绝大多数模型仅直接继承或微调自少量上游基座模型, 2–3 跳链条数量的快速下降反映了多数模型家族的层级较浅; 另一方面, 在 5–10 跳区间, 路径数量重新上升, 可能存在少数具有多级派生和丰富分支结构的深度模型家族. 沿着这些主干链条向下游展开时, 中间节点被多个长路径重复经过, 分支组合也会在更大跳数上成倍叠加, 从而使长链路径数量呈现出非线性增长. 从功能角度看, 这类多级链条体现了知识迁移和迭代复用, 有助于在现有模型之上不断派生新变体, 并不必然等同于负面现象; 但从结构性风险视角来看, 一旦上游基座模型存在严重缺陷, 其影响就可能沿着这些深层依赖链在有限拓扑距离内传递至大量下游模型, 显著放大故障的传播半径和影响范围.

综合来看, 前几跳统计主要由大量层级较浅的小型模型家族主导, 因而路径数随跳数增加快速衰减; 而在更高跳数段, 少数具有深层多级派生和复杂分支结构的模型家族开始主导统计, 它们通过中间节点的重复经过和分支组合效应, 推动长链路径数重新上升, 从而形成“先降低、后上升”的非单调趋势.

与之形成对比的是, 数据集依赖链 source(extend)_data_from 在图 9 的半对数坐标下大致呈近似线性的单调下降, 即路径数量随跳数增长近似指数衰减, 最大有效依赖深度约为 4 跳. 该模式表明, 数据集之间的交叉引用整体较为有限: 大多数数据复用以“浅层 1 跳”形式出现, 即直接引用少数上游基础数据集, 而很少进一步在下游派生出多级级联的“数据家族”. 同时, 数据集往往围绕特定任务或场景独立设计, 这也在一定程度上抑制了类似模型那样

的多级派生链条的形成, 使得跨数据集的依赖深度整体受到限制.



(a) 线性坐标 (b) 对数坐标

图 9 base_model 与 source(extend)_data_from 在不同跳数下的路径数量变化

需要强调的是, 模型依赖链条的“更长”与数据集依赖链条的“更短”, 其形成机制和实际含义并不完全等价. 模型链条主要刻画的是参数与表示在多代模型中的迁移与累积, 长链在促进模型创新和能力扩展的同时, 也为错误与偏差提供了更长的传播通道; 数据集链条则更多反映数据衍生、清洗与任务关联的关系, 较浅的链条一方面降低了跨数据集级联失效的风险, 另一方面也提示当前生态中跨任务、跨领域的数据整合仍然有限.

因此, 本文将链条长度视为结构性风险在“路径结构”维度上的一个关键刻画: 在模型依赖网络中, 少数深度派生家族与高度异质的节点度分布叠加, 构成了局部异常得以在多跳路径上放大和扩散的主要通道; 相较之下, 数据集依赖链的整体扁平结构在一定程度上约束了风险传播的深度, 但也暴露出数据共享和复用模式的局限性.

发现 3: 模型的 base_model 依赖呈现“前浅后深”结构, 大部分模型链条较浅, 但少数深度派生家族在高跳数段形成大量长链, 使上游错误能够被多跳放大并快速传递. 相比之下, 数据集依赖链整体较浅、随跳数指数衰减, 风险传播深度有限, 但也反映了跨任务数据复用不足.

(3) 演化特征

如图 10 所示, 模型、数据集与作者数量在观察区间内均呈持续增长趋势, 但增长幅度存在显著不均衡. 模型数量由约 3 万迅速上升至超过 57 万, 呈现爆炸式扩张; 数据集规模从约 0.35 万增长至 12 万左右; 而独立作者的累计数量则从 9.5 万增长到约 17 万, 始终显著低于资源规模. 这表明, 生态的扩张主要由模型端驱动, 而并非由新数据集和新作者推动.

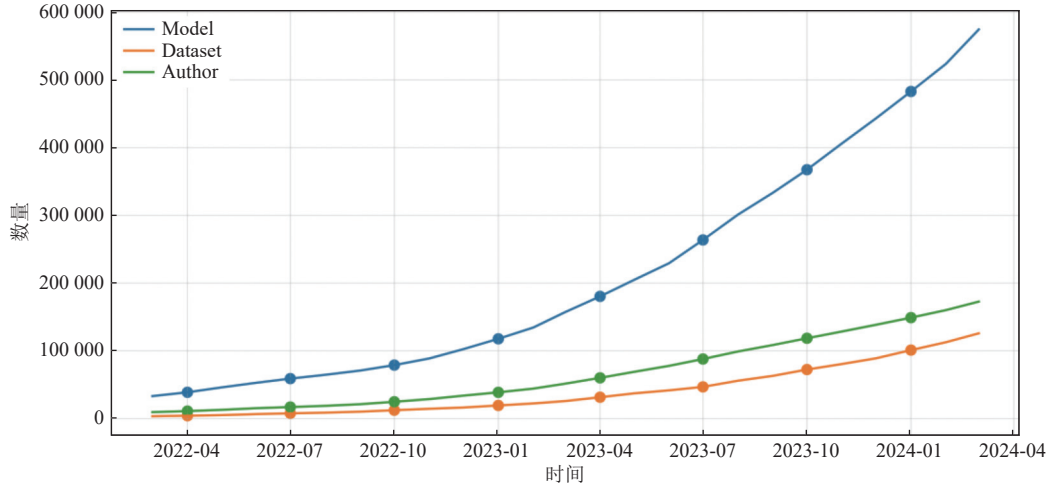


图 10 模型 (Model)、数据集 (Dataset)、作者 (Author) 数量随时间变化曲线

对关键依赖关系的进一步分析 (见图 11) 加深了我们对生态系统内部结构性依赖的理解. 其中, 与数据集相关的依赖关系 source_data_from 和 extend_data_from 在数量上始终保持较低水平, 且增长缓慢. 该趋势表明, 生态系统中数据集的扩展与引用较为有限.

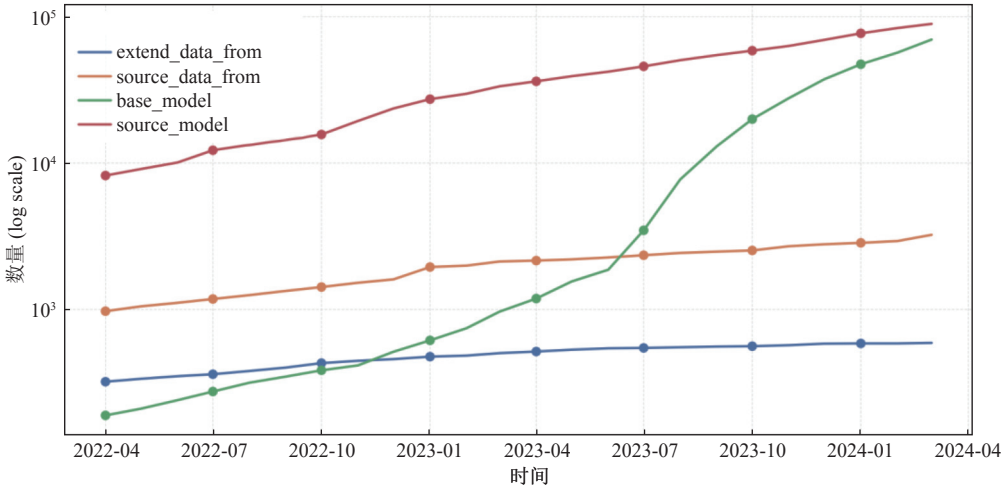


图 11 主要关系数量随时间变化曲线

相比之下, 模型与数据集之间的依赖关系 source_model 呈现出显著增长趋势, 其数量从 2022 年 4 月的不足 10^4 条, 到 2024 年初已逼近 10^5 条. 这一趋势表明, 模型的开发与复用主要依赖于现有数据集, 而非推动新数据集的创建. 随着模型数量的快速增长而数据集扩展相对滞后, 生态系统可能面临对有限数据集过度依赖的风险. 该依赖关系的不断强化不仅可能加剧数据瓶颈问题, 还可能在长期演化中促使数据偏倚的积累[55].

在作者维度上, 如图 12 所示, 每月新增的资源 (模型与数据集) 数量始终显著高于新增作者数量, 两者大致保持在 2–4 倍的差距; 累积作者规模也始终落后于资源规模, 说明维护主体的扩张速度无法跟上资源生产速度, 生态呈现出“资源增速远快于作者增速”的典型不平衡模式.

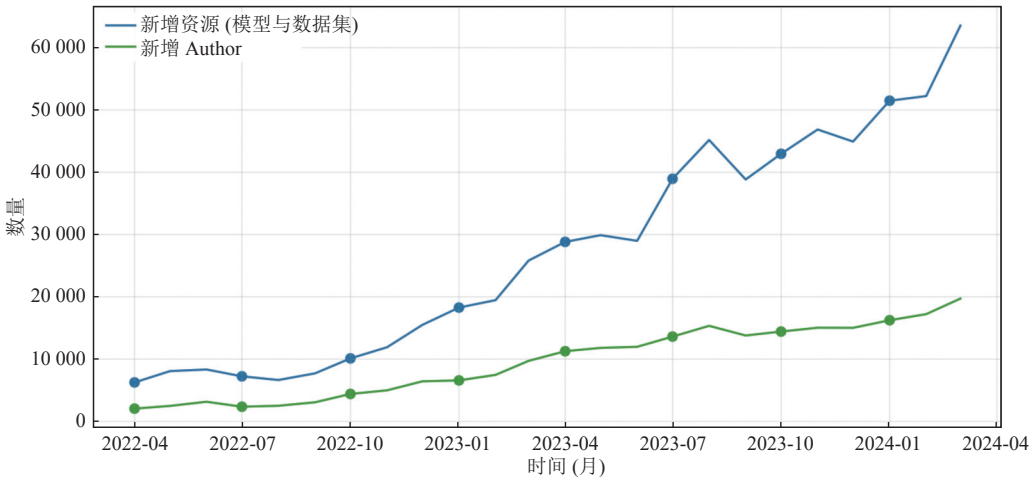


图 12 新增资源 (模型与数据集) 数量与新增作者数量的时间变化趋势

综合上述结果可以看出: HF 模型生态在时间维度上呈现出显著的不均衡演化特征, 即模型端高速扩张, 而数据集与作者端相对滞后. 这种动态失衡使得依赖网络逐步强化对少数核心数据集与有限作者群体的结构性依赖,

从而使系统在长期演化过程中呈现出不断累积的系统性风险. 该结论与前文关于节点度分布与多跳依赖链的静态分析相互印证, 说明少数高连接节点和深层链条在演化过程中持续成为潜在风险的传播中心.

发现 4: 模型数量的爆炸式增长远远超过数据集扩展和作者群体的增长, 关键依赖关系也持续向少数数据源集中. 随着这种不均衡不断累积, 生态对有限数据集与有限作者的长期路径依赖被持续强化, 使得潜在错误、偏倚或过时信息更易在不断扩大的依赖链条中被放大.

3.2 RQ2: 如何量化 AI 资源依赖带来的可信性风险?

3.2.1 指标有效性分析

(1) 参数稳健性

参数敏感性实验结果如表 5 和表 6 所示. 无论在模型侧还是数据集侧, 对关键参数 α (指数惩罚项的放大系数)、β (对行为变量进行平滑处理的对数项系数)、γ (用于 out-degree 平滑) 施加±20% 扰动后, 所有情形下的 Spearman 排序相关系数均保持在 0.999 以上; 而风险最高与最低的前 10% 节点的重合率在大多数情况下达到 1.0.

表 5 模型节点的参数敏感性分析结果

参数	扰动	Spearman相关系数	Top-10%重合率
α	0.8	0.999910	1.000
	1.2	0.999926	1.000
β	0.8	0.999994	1.000
	1.2	0.999995	1.000
γ	0.8	0.999993	1.000
	1.2	0.999998	1.000

表 6 数据集节点的参数敏感性分析结果

参数	扰动	Spearman相关系数	Top-10%重合率
α	0.8	0.999509	0.999104
	1.2	0.999951	0.999925
β	0.8	0.999997	1.000
	1.2	0.999995	1.000
γ	0.8	0.999998	1.000
	1.2	0.999998	1.000

这表明整体排序结构几乎不受参数变化影响, 关键风险节点也在不同参数设定下保持高度一致.

(2) 基于结构化正负样本的判别能力验证

在全部 7 个风险维度上, 负样本均表现出更高的风险水平 (见图 13), 各维度的差异均达到统计学显著性 (p<0.05, 其中大部分维度达到 p<0.001), 表明所提出的风险指标能够在不同风险来源下稳定捕捉结构性差异. 效应量分析进一步提供了一致的分布层面证据 (见表 7), 从而佐证该指标具有良好的区分能力.

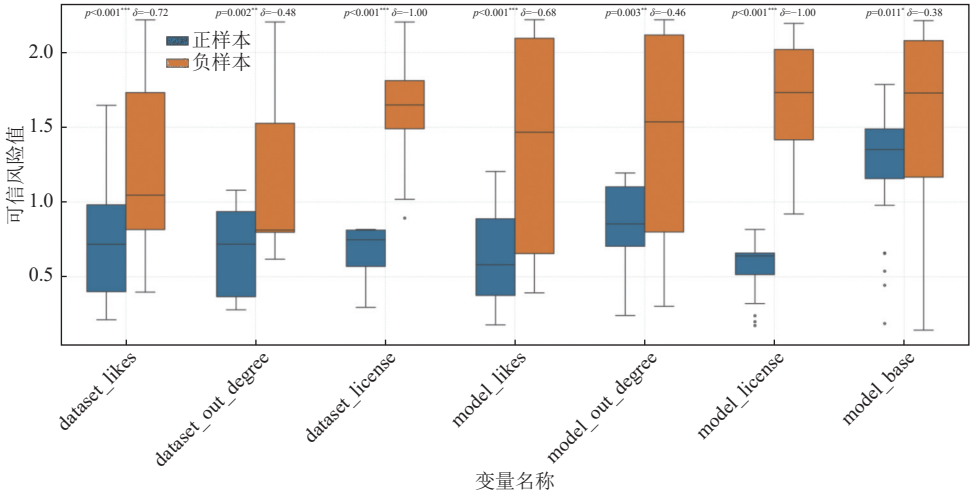


图 13 7 个风险维度的风险评分比较

(3) 基线方法对比

与 license-only 基线相比, 仅依赖许可证/作者缺失所形成的离散分层难以支持细粒度排序, 导致大量节点风险值并列; 而可信风险指标输出连续排序, 可进一步区分同一合规类别内部的风险差异.

表 7　7 个风险维度的显著性检验结果

Task	p	Effect size (δ)
dataset_likes	<0.001 (***)	−0.72 (大效应)
dataset_out_degree	0.002 (**)	−0.48 (大效应)
dataset_license	<0.001 (***)	−1.00 (大效应)
model_likes	<0.001 (***)	−0.68 (大效应)
model_out_degree	0.003 (**)	−0.46 (中等效应)
model_license	<0.001 (***)	−1.00 (大效应)
model_base	0.011 (*)	−0.38 (中等效应)

注: 显著性标记: *表示p<0.05; **表示p<0.01; ***表示p<0.001

与 feedback-only 基线相比, 两者的 Top-20 风险排名重合度在模型侧为 0.70、数据集侧为 0.55, 均显著低于 1, 说明二者识别出的高风险头部并不一致. 该结果表明: 社区反馈相关信号能够反映部分风险, 但不足以覆盖可信风险指标所捕捉的多因素复合风险; 完整指标在排序与风险识别上提供了额外、不可替代的信息增益.

(4) 专家评审

专家盲评结果表明, 在高风险子集中, 专家判断与指标输出在 8 个节点上完全一致, 另有 2 个节点存在偏差. 经事后分析发现, 偏差主要源于跨平台更新延迟, 其中部分节点依赖的多源数据未能及时同步至主流平台 (如 HF), 导致在外部信息缺失的情况下指标对其风险水平产生一定高估. 相比之下, 在低风险子集中, 专家对节点风险较低的判断与指标排序保持一致, 并在可解释性与实用性方面给予较高评价, 说明该指标在风险样本上具有良好的判别能力.

总体来看, 专家反馈普遍肯定了该指标的可理解性与操作性, 认为其特别适用于大规模依赖网络中的快速预筛与风险预警. 尽管在少数节点上存在因跨平台信息不同步导致的偏差, 但专家一致认为该问题并未影响研究的核心结论, 亦不构成对结果有效性的实质性威胁. 这些发现表明, 可信风险指标在当前框架下展现出较高的实用价值与应用前景.

3.2.2　节点风险分析

(1) 风险耦合分析

风险并非孤立分布. 我们首先将模型-数据集显式依赖按“(高/低风险模型)×(高/低风险数据集”) 划分, 得到了 2×2 配对表, 如表 8 所示.

表 8　模型端与数据集端的风险等级组合情况 (2×2 配对表) (条)

类别	数量
高风险模型-高风险数据集	360
高风险模型-低风险数据集	1 027
低风险模型-高风险数据集	176
低风险模型-低风险数据集	16 834

对该配对表进行卡方独立性检验得到 χ^2=2806.8, p<0.001, 说明显著拒绝独立性假设, 在高风险模型的所有显式依赖中, 约有 26.0% 指向高风险数据集; 而在低风险模型的依赖中, 仅有约 1.0% 指向高风险数据集. 对于每个数据集, 计算其自身的风险得分以及所有下游模型风险得分的平均值, 并使用 Spearman 等级相关系数进行统计检验. 结果表明, 数据集风险与其下游模型平均风险之间的相关系数 ρ 约为 0.61 (p<0.001), 属于中-高强度的正相关关系. 表明二者存在中等偏强的正相关关系, 说明高风险数据集更倾向于与平均风险较高的模型共同出现, 从而形成局部的风险耦合结构.

(2) 任务维度的风险分析

将风险映射到任务维度后, 不同任务间的平均风险存在明显差异, 且模型端与数据集端呈非对称表现. 具体分布情况如图 14 所示.

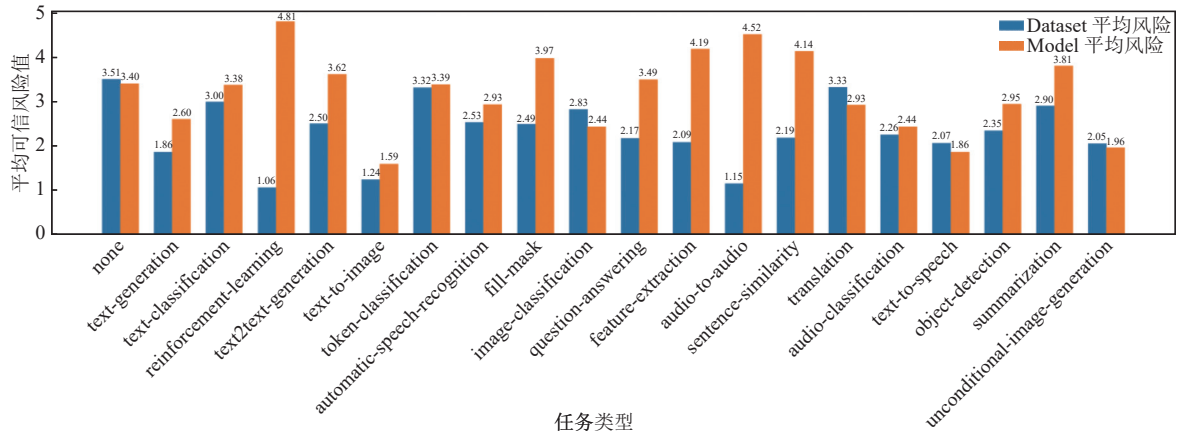


图 14　不同任务类别下资源 (Dataset 和 Model) 平均可信风险的对比分析

在模型侧, reinforcement-learning 任务的模型平均风险约为 4.81, 在所有任务中最高; fill-mask、feature-extraction、audio-to-audio、sentence-similarity 等任务的模型平均风险也普遍接近或超过 4. 相较之下, text-generation 与 image-classification 的模型平均风险处于中间区间, 分别约为 2.60 和 2.44, 而 text-to-image 的模型平均风险相对较低, 仅约 1.59.

在数据集侧, 风险分布更加分散, 且与模型端的排序并不一致: none 类别的数据集平均风险约为 3.51, 高于大多数具有明确任务标注的类别; text-classification、token-classification 和 fill-mask 的数据集平均风险分别约为 3.00、3.32、2.49; automatic-speech-recognition 与 image-classification 的数据集平均风险约为 2.53 和 2.83; 而 text-generation、text-to-image 与 reinforcement-learning 的数据集平均风险分别约为 1.86、1.24、1.06, 整体处于较低水平. 与其模型端风险形成对照, 提示不同端侧的风险来源权重可能存在差异.

进一步基于全局分位阈值进行高/低风险分层后, 任务间还呈现尾部分布强度的显著差别 (见图 15). 具体来说, 部分任务虽然风险均值处于中等区间, 但其高风险占比明显偏高, 表明风险主要集中在少量尾部节点; 反之, 也存在均值偏高但高风险占比未同比例升高的任务. 此外, 模型端与数据集端的尾部表现仍然不对称: 总体上, 模型侧的高风险占比在多个任务中显著高于数据集侧, 而数据集侧在若干任务中保持较低的高风险比例. 这一结果进一步说明, 任务差异不仅体现在平均风险水平, 还体现在风险分布的偏斜程度与尾部厚度, 即不同任务对“高风险节点”的聚集强度不同.

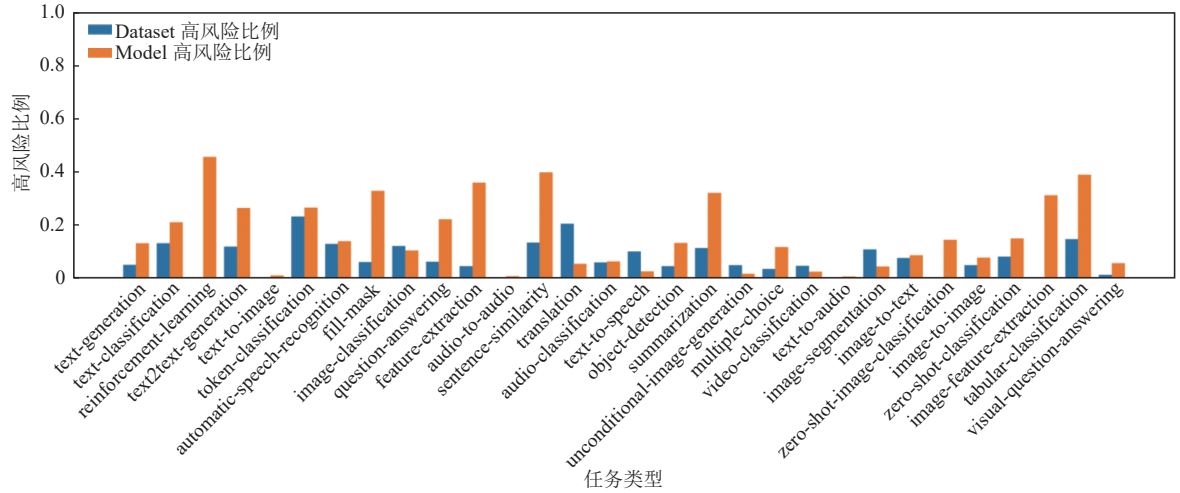


图 15　不同任务类型的高风险比例

(3) 典型高/低风险节点识别

基于风险排序, 我们选取了可信风险最高的 5 个数据集与 5 个模型, 以及可信风险最低的 5 个数据集与 5 个模型, 如表 9 与表 10 所示, 并对影响可信度的共性特征进行了分析. 完整的全局排序结果 (覆盖所有模型与数据集的可信风险得分) 已随复现包一并提供.

表 9　数据集的可信风险评估

类型	数据集	作者	许可证类型	创建时间	出度	入度	点赞数	可信风险值
可信风险最高的5个数据集	ajgt_twitter_ar	none	unknown	2022-03-02	5	0	3	20.266
	aquamuse	none	unknown	2022-03-02	12	0	9	20.266
	ar_res_reviews	none	unknown	2022-03-02	5	0	5	20.266
	biomrc	none	none	2022-03-02	1	0	4	20.266
	bookcorpusopen	none	unknown	2022-03-02	9	0	30	20.266
可信风险最低的5个数据集	LDJnr/Capybara	LDJnr	Apache-2.0	2023-12-16	3	364	185	0.179
	fka/awesome-chatgpt-prompts	fka	cc0-1.0	2022-12-13	1	451	4 873	0.360
	Open-Orca/OpenOrca	Open-Orca	mit	2023-06-15	12	534	1 149	0.285
	mozilla-foundation/common_voice_11_0	mozilla-foundation	cc0-1.0	2022-10-12	5	923	143	0.339
	mozilla-foundation/common_voice_7_0	mozilla-foundation	cc0-1.0	2022-03-02	5	1 325	23	0.366

表 10　模型的可信风险评估

类型	模型	作者	许可证类型	创建时间	出度	入度	点赞数	可信风险值
可信风险最高的5个模型	google-bert/bert-base-cased-finetuned-mrpc	google-bert	none	2022-03-02	1	0	1	7.455
	transfo-xl/transfo-xl-wt103	transfo-xl	none	2022-03-02	3	0	9	7.455
	FacebookAI/xlm-clm-ende-1024	FacebookAI	none	2022-03-02	3	0	0	7.455
	FacebookAI/xlm-clm-enfr-1024	FacebookAI	none	2022-03-02	3	0	0	7.455
	FacebookAI/xlm-roberta-large-finetuned-conll02-dutch	FacebookAI	none	2022-03-02	3	0	1	7.455
可信风险最低的5个模型	google/vit-base-patch16-224-in21k	Google	Apache-2.0	2023-03-02	4	859	127	0.388
	openai/whisper-small	OpenAI	Apache-2.0	2022-09-26	2	875	144	0.346
	openai/whisper-tiny	OpenAI	Apache-2.0	2022-09-26	2	944	167	0.342
	mistralai/Mistral-7B-v0.1	MistralAI	Apache-2.0	2023-09-20	2	1 524	3 066	0.203
	stabilityai/stable-diffusion-xl-base-1.0	StabilityAI	OpenRAIL++	2023-07-25	5	3 391	4 890	0.208

对前 5 个高风险数据集的分析表明, 其普遍存在关键信息缺失的模式, 尤其是在许可证信息 (常被标记为 unknown 或 none) 和作者标注 (none) 方面. 这些核心元数据的缺失使得数据来源、使用权限及维护责任的确认变得复杂. 此外, 这些数据集在生态系统中缺乏活跃的复用与引用, 表现为几乎没有下游依赖.

同样, 排名前 5 的高风险模型也普遍存在许可证信息缺失的问题. 需要注意的是, 这类问题并非仅出现在冷门模型中, 像 google-bert/bert-base-cased-finetuned-mrpc、FacebookAI/xlm-clm-ende-1024 等较常见的模型, 也可能因缺乏明确的授权许可文档而带来潜在的合规风险. 许可证信息的不明确将导致其可使用范围存在不确定性, 并对下游的采纳与复用造成限制. 需要说明的是, 高风险 Top-5 节点风险值相同, 源于其共同满足“许可证/作者关键缺失、低社区反馈等”的一致条件, 导致指数惩罚主导且其余项差异不足以拉开分值.

相比之下, 低风险的模型与数据集通常具备明确的许可证与作者标注, 更有可能被下游模型作为依赖项使用, 显示其在生态系统中具有更广泛的影响力与信任度. 它们往往还获得更多社区“点赞”, 反映出良好的用户认可度与持续维护性. 这一对比验证了作者信息与许可证完整性在可信风险评估中的核心作用, 强调了透明授权与责任归属对开源生态健康发展的重要性.

在定量排名的基础上, 我们进一步纳入了定性案例分析, 以更直观地展示指标在真实情境中的实用价值. 以处于高风险的 google-bert/bert-base-chinese 模型为例, 该模型虽在社区中具有一定使用可见度, 但其仓库未声明任何许可证. 根据本文的指标设计, 许可证属于灾难敏感型的关键合规元数据, 一旦缺失将触发指数惩罚并显著抬升总体风险. 与此同时, 根据追溯发现, 这一缺陷导致下游模型在继承与传播该模型时出现许可标注混乱, 例如部分被

误标为 MIT 或 Apache-2.0. 该案例表明, 即便节点在其他维度可能具有一定的社区可用性或外部验证基础, 关键合规信息的缺失仍会以门槛效应主导其可信风险水平, 使其成为需要优先治理的潜在高危资源, 从而验证了“可信风险”框架在识别此类节点方面的有效性.

4 启 示

4.1.1 对开发者的启示

开发者应高度重视模型与数据集元数据的完整性, 尤其是许可证和作者信息的准确性. 这些关键信息的缺失不仅会影响资源的合规性, 还可能引发版权纠纷. 此外, 为降低系统性风险的累积, 开发者应积极推动依赖关系的多样化, 避免过度依赖单一来源的数据集或模型. 在数据处理方面, 应注重数据标注的标准化和透明化, 提升标注质量, 从源头减少因偏差而引发的潜在风险.

4.1.2 对研究者的启示

本研究发现, 网络中高度中心化的现象不仅反映了某些模型或数据集在平台上的广泛应用, 也暴露出潜在的系统性脆弱性. 这种现象可能源于对关键模型或数据集的过度依赖, 尽管这种做法在学术界中被广泛采用以提升研究表现[56]. 然而, Crawford 等人[57]与 Birhane 等人[58]警示称, 这种依赖关系可能导致贬义标签或冒犯性内容的传播, 从而引发严重的社会文化问题. 此外, Geirhos 等人[59]和 D’Amour 等人[60]亦指出, 许多评估基准和指标在真实世界场景中的有效性尚未得到充分验证, 暴露出现有方法在应对复杂性方面的不足. 因此, 研究者应警惕这种依赖关系可能带来的伦理和社会文化问题, 并在模型开发和数据集使用过程中保持审慎态度.

4.1.3 对平台服务提供者的启示

平台应优先完善许可证和作者信息的记录与校验机制, 建立标准化的元数据框架, 涵盖授权条款、作者归属、版本演化和使用限制等关键信息, 从而弥补当前模型卡和数据集卡在信息表达上的不足[31,37]. 在此基础上, 平台还应积极引入跨平台数据采集与整合机制, 以缓解因多源数据更新延迟或不同步所造成的风险高估现象, 从而提升对模型与数据集全生命周期的感知与监测能力. 同时, 平台应将“可信风险”指标嵌入持续集成/持续部署 (CI/CD) 流程, 自动阻断高风险版本, 并提示补充缺失的元数据. 此外, 平台可以借鉴 StackOverflow 等平台的激励机制 (如积分、徽章、等级等)[61], 鼓励社区积极参与资源更新与维护, 避免高价值资源因缺乏维护而陷入“沉寂”, 从而提升平台的长期活跃度. 同时, 平台应推动数据标注工作的激励机制, 鼓励更多贡献者参与数据集与模型的标注任务, 并促进数据集和模型之间依赖关系的多样化. 这将有助于降低对少数核心节点的过度依赖, 从而减少系统性风险的产生. 最后, 平台应结合本研究提出的“可信风险”指标, 构建实时风险监测仪表盘, 并根据指标反馈对高风险资源进行自动预警. 该仪表盘能够帮助开发者和研究者及时识别潜在高风险模型和数据集, 并采取相应应对措施.

5 总 结

本研究通过对 HF 平台的大规模实证分析, 系统揭示了开源模型生态中资源依赖的结构特征与风险分布. 研究发现, 该生态在高速扩张的同时, 呈现出显著的结构性失衡: 依赖关系高度集中于少数枢纽节点, 且数据集侧的来源追溯问题尤为突出, 导致生态整体面临单点故障与传播链脆弱性风险. 演化分析进一步表明, 模型数量的爆炸式增长加剧了对核心数据集的路径依赖, 系统性风险在持续累积. 为应对上述挑战, 本文提出了一个融合多维度信号的“可信风险”量化指标. 该指标通过综合考量节点的属性完整性、社区反馈, 有效识别生态中的关键风险节点, 并在多个维度的验证中展现出良好的稳健性与判别力. 总体而言, 本文从生态结构与节点风险两个层面, 为理解开源模型生态的依赖治理提供了系统性的分析框架与实证依据. 所构建的“可信风险”指标为平台级资源筛查与风险预警提供了可操作的工具支持. 为缓解这些风险, 我们建议提升元数据标准, 并推动数据集多样化, 增强生态系统的可持续性. 未来的研究将进一步完善“可信风险”评估框架, 并拓展数据来源, 以提升其适用性与评估精度.

References

[1] Pepe F, Nardone V, Mastropaolo A, Bavota G, Canfora G, Di Penta M. How do Hugging Face models document datasets, bias, and

licenses? An empirical study. In: Proc. of the 32nd IEEE/ACM Int'l Conf. on Program Comprehension. Lisbon: ACM, 2024. 370–381. [doi: 10.1145/3643916.3644412]

[2] Zewe A. Study: Transparency is often lacking in datasets used to train large language models. 2024. https://dspace.mit.edu/handle/1721.1/157453

[3] He RZ, Zhou MH. Countermeasures for software supply chain risks of the intelligent era. Computing Magazine of the CCF, 2025, 1(3): 59–66 (in Chinese with English abstract). [doi: 10.11991/cccf.202507009]

[4] Mora-Cantallops M, Sánchez-Alonso S, García-Barriocanal E. A complex network analysis of the comprehensive R archive network (CRAN) package ecosystem. Journal of Systems and Software, 2020, 170: 110744. [doi: 10.1016/j.jss.2020.110744]

[5] Kikas R, Gousios G, Dumas M, Pfahl D. Structure and evolution of package dependency networks. In: Proc. of the 14th IEEE/ACM Int'l Conf. on Mining Software Repositories. Buenos Aires: IEEE, 2017. 102–112. [doi: 10.1109/msr.2017.55]

[6] Hu Y, Zhang J, Bai XM, Yu S, Yang Z. Influence analysis of GitHub repositories. SpringerPlus, 2016, 5(1): 1268. [doi: 10.1186/s40064-016-2897-7]

[7] Thung F, Bissyandé TF, Lo D, Jiang LX. Network structure of social coding in GitHub. In: Proc. of the 17th European Conf. on Software Maintenance and Reengineering. Genova: IEEE, 2013. 323–326. [doi: 10.1109/csmr.2013.41]

[8] Jiang J, Zhang L, Li L. Understanding project dissemination on a social coding site. In: Proc. of the 20th Working Conf. on Reverse Engineering. Koblenz: IEEE, 2013. 132–141. [doi: 10.1109/wcre.2013.6671288]

[9] Vasilescu B, Serebrenik A, Filkov V. A data set for social diversity studies of GitHub teams. In: Proc. of the 12th IEEE/ACM Working Conf. on Mining Software Repositories. Florence: IEEE, 2015. 514–517. [doi: 10.1109/msr.2015.77]

[10] Yu Y, Yin G, Wang HM, Wang T. Exploring the patterns of social behavior in GitHub. In: Proc. of the 1st Int'l Workshop on Crowd-based Software Development Methods and Technologies. Hong Kong: ACM, 2014. 31–36. [doi: 10.1145/2666539.2666571]

[11] Decan A, Mens T, Grosjean P. An empirical comparison of dependency network evolution in seven software packaging ecosystems. Empirical Software Engineering, 2019, 24(1): 381–416. [doi: 10.1007/s10664-017-9589-y]

[12] Valiev M, Vasilescu B, Herbsleb J. Ecosystem-level determinants of sustained activity in open-source projects: A case study of the PyPI ecosystem. In: Proc. of the 26th ACM Joint Meeting on European Software Engineering Conf. and Symp. on the Foundations of Software Engineering. Lake Buena: ACM, 2018. 644–655. [doi: 10.1145/3236024.3236062]

[13] Ji SL, Wang QY, Chen AY, Zhao BB, Ye T, Zhang XH, Wu JZ, Li Y, Yin JW, Wu YJ. Survey on open-source software supply chain security. Ruan Jian Xue Bao/Journal of Software, 2023, 34(3): 1330–1364 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6717.htm [doi: 10.13328/j.cnki.jos.006717]

[14] Gao K, He H, Xie B, Zhou MH. Survey on open source software supply chains. Ruan Jian Xue Bao/Journal of Software, 2024, 35(2): 581–603 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6975.htm [doi: 10.13328/j.cnki.jos.006975]

[15] Kathikar A, Nair A, Lazarine B, Sachdeva A, Samtani S. Assessing the vulnerabilities of the open-source artificial intelligence (AI) landscape: A large-scale analysis of the Hugging Face platform. In: Proc. of the 2023 IEEE Int'l Conf. on Intelligence and Security Informatics. Charlotte: IEEE, 2023. 1–6. [doi: 10.1109/isi58743.2023.10297271]

[16] Castaño J, Martínez-Fernández S, Franch X, Bogner J. Exploring the carbon footprint of Hugging Face's ML models: A repository mining study. In: Proc. of the 2023 ACM/IEEE Int'l Symp. on Empirical Software Engineering and Measurement. New Orleans: IEEE, 2023. 1–12. [doi: 10.1109/esem56168.2023.10304801]

[17] Jiang WX, Synovic N, Hyatt M, Schorlemmer TR, Sethi R, Lu YH, Thiruvathukal GK, Davis JC. An empirical study of pre-trained model reuse in the Hugging Face deep learning model registry. In: Proc. of the 45th IEEE/ACM Int'l Conf. on Software Engineering. Melbourne: IEEE, 2023. 2463–2475. [doi: 10.1109/icse48619.2023.00206]

[18] Castaño J, Martínez-Fernández S, Franch X, Bogner J. Analyzing the evolution and maintenance of ML models on Hugging Face. In: Proc. of the 21st Int'l Conf. on Mining Software Repositories. Lisbon: ACM, 2024. 607–618. [doi: 10.1145/3643991.3644898]

[19] Katzy J, Popescu R, van Deursen A, Izadi M. An exploratory investigation into code license infringements in large language model training datasets. In: Proc. of the 1st IEEE/ACM Int'l Conf. on AI Foundation Models and Software Engineering. Lisbon: ACM, 2024. 74–85. [doi: 10.1145/3650105.3652298]

[20] Ma J, Zhou K, Ren Y, Zhu H, Qin Y, Wang J. Package dependency analysis for operating system distribution building. Computer Engineering and Science, 2021, 43(11): 1926–1933 (in Chinese with English abstract). [doi: 10.3969/j.issn.1007-130X.2021.11.004]

[21] Wang ZB, Jia XK, Ying LY, Su PR. Fine-grained assessment method of vulnerability impact scope for PyPI ecosystem. Ruan Jian Xue Bao/Journal of Software, 2024, 35(10): 4493–4509 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/6959.htm [doi: 10.13328/j.cnki.jos.006959]

[22] Rahman MS, Gao P, Ji YD. HuggingGraph: Understanding the supply chain of LLM ecosystem. In: Proc. of the 34th ACM Int'l Conf. on Information and Knowledge Management. Seoul: ACM, 2025. 5997–6005. [doi: 10.1145/3746252.3761510]

[23] Stalnaker T, Wintersgill N, Chaparro O, Heymann LA, Di Penta M, German DM, Poshyvanyk D. An empirical analysis of machine

learning model and dataset documentation, supply chain, and licensing challenges on Hugging Face. ACM Trans. on Software Engineering and Methodology, 2025. [doi: 10.1145/3776739]

[24] Yang Z, Shi JK, Devanbu P, Lo D. Ecosystem of large language models for code. ACM Trans. on Software Engineering and Methodology, 2025, 35(1): 12. [doi: 10.1145/3731753]

[25] Haq EU, Wang S, Allison RS. The ripple effect of vulnerabilities in Maven Central: Prevalence, propagation, and mitigation challenges. In: Proc. of the 22nd IEEE/ACM Int'l Conf. on Mining Software Repositories. Ottawa: IEEE, 2025. 344–348. [doi: 10.1109/MSR66628.2025.00063]

[26] Moukafih N, Zhang HS, Epiphaniou G, Maple C, Taylor S, Carmichael L. Semi-automated threat vulnerability & risk assessment (TVRA) for medical devices. In: Proc. of the 17th Int'l Conf. on PErvasive Technologies Related to Assistive Environments. Crete: ACM, 2024. 687–693. [doi: 10.1145/3652037.3663896]

[27] Ramaj X, Sánchez-Gordón M, Gkioulos V, Colomo-Palacios R. On DevSecOps and risk management in critical infrastructures: Practitioners' insights on needs and goals. In: Proc. of the 2024 ACM/IEEE Int'l Workshop on Engineering and Cybersecurity of Critical Systems (EnCyCriS) and 2nd IEEE/ACM Int'l Workshop on Software Vulnerability. Lisbon: ACM, 2024. 45–52. [doi: 10.1145/3643662.3643954]

[28] Gebru T, Morgenstern J, Vecchione B, Vaughan JW, Wallach H, Daumé III H, Crawford K. Datasheets for datasets. Communications of the ACM, 2021, 64(12): 86–92. [doi: 10.1145/3458723]

[29] Mitchell M, Wu S, Zaldivar A, Barnes P, Vasserman L, Hutchinson B, Spitzer E, Raji ID, Gebru T. Model cards for model reporting. In: Proc. of the 2019 Conf. on Fairness, Accountability, and Transparency. Atlanta: ACM, 2019. 220–229. [doi: 10.1145/3287560.3287596]

[30] Osborne C, Ding J, Kirk HR. The AI community building the future? A quantitative analysis of development activity on Hugging Face Hub. Journal of Computational Social Science, 2024, 7(2): 2067–2105. [doi: 10.1007/s42001-024-00300-8]

[31] Yang XY, Liang WX, Zou J. Navigating dataset documentations in AI: A large-scale analysis of dataset cards on HuggingFace. In: Proc. of the 12th Int'l Conf. on Learning Representations. Vienna: OpenReview.net, 2024.

[32] Schwabe D, Becker K, Seyferth M, Klaß A, Schaeffter T. The METRIC-framework for assessing data quality for trustworthy AI in medicine: A systematic review. NPJ Digital Medicine, 2024, 7(1): 203. [doi: 10.1038/s41746-024-01196-4]

[33] Schlegel M, Sattler KU. Capturing end-to-end provenance for machine learning pipelines. Information Systems, 2025, 132: 102495. [doi: 10.1016/j.is.2024.102495]

[34] Taraghi M, Dorcelus G, Foundjem A, Tambon F, Khomh F. Deep learning model reuse in the HuggingFace community: Challenges, benefit and trends. In: Proc. of the 2024 IEEE Int'l Conf. on Software Analysis, Evolution and Reengineering. Rovaniemi: IEEE, 2024. 512–523. [doi: 10.1109/saner60148.2024.00059]

[35] Ait A, Cánovas Izquierdo JL, Cabot J. HFCommunity: An extraction process and relational database to analyze Hugging Face Hub data. Science of Computer Programming, 2024, 234: 103079. [doi: 10.1016/j.scico.2024.103079]

[36] Guia J, Soares VG, Bernardino J. Graph databases: Neo4j analysis. In: Proc. of the 19th Int'l Conf. on Enterprise Information Systems. Porto: ICEIS, 2017. 351–356.

[37] Liang WX, Rajani N, Yang XY, Ozoani E, Wu E, Chen YQ, Smith DS, Zou J. Systematic analysis of 32 111 AI model cards characterizes documentation practice in AI. Nature Machine Intelligence, 2024, 6(7): 744–753. [doi: 10.1038/s42256-024-00857-z]

[38] Newman MEJ. The structure and function of complex networks. SIAM Review, 2003, 45(2): 167–256. [doi: 10.1137/S003614450342480]

[39] Alstott J, Bullmore E, Plenz D. powerlaw: A Python package for analysis of heavy-tailed distributions. PLoS One, 2014, 9(1): e85777. [doi: 10.1371/journal.pone.0085777]

[40] Wu JQ, Bao LF, Yang XH, Xia X, Hu X. A large-scale empirical study of open source license usage: Practices and challenges. In: Proc. of the 21st Int'l Conf. on Mining Software Repositories. Lisbon: ACM, 2024. 595–606. [doi: 10.1145/3643991.3644900]

[41] Kaplan AD, Kessler TT, Brill JC, Hancock PA. Trust in artificial intelligence: Meta-analytic findings. Human factors, 2023, 65(2): 337–359. [doi: 10.1177/00187208211013988]

[42] Acerbi C. Spectral measures of risk: A coherent representation of subjective risk aversion. Journal of Banking & Finance, 2002, 26(7): 1505–1518. [doi: 10.1016/s0378-4266(02)00281-9]

[43] Dowd K, Cotter J, Sorwar G. Spectral risk measures: Properties and limitations. Journal of Financial Services Research, 2008, 34(1): 61–75. [doi: 10.1007/s10693-008-0035-6]

[44] Celma Ò. The long tail in recommender systems. In: Music Recommendation and Discovery: The Long Tail, Long Fail, and Long Play in the Digital Music Space. Berlin Heidelberg: Springer, 2010. 87–107. [doi: 10.1007/978-3-642-13287-2_4]

[45] Carnovalini F, Rodà A, Wiggins GA. Popularity bias in recommender systems: The search for fairness in the long tail. Information, 2025, 16(2): 151. [doi: 10.3390/info16020151]

[46] Wu F, Huberman BA. Novelty and collective attention. Proc. of the National Academy of Sciences of the United States of America, 2007,

104(45): 17599–17601. [doi: 10.1073/pnas.0704916104]
[47] de Winter JCF, Gosling SD, Potter J. Comparing the Pearson and Spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. Psychological Methods, 2016, 21(3): 273. [doi: 10.1037/met0000079]
[48] McKnight PE, Najab J. Mann-Whitney U test. In: Weiner IB, Edward Craighead W, eds. The Corsini Encyclopedia of Psychology. Hoboken: John Wiley, 2010. 1. [doi: 10.1002/9780470479216.corpsy0524]
[49] Meissel K, Yao ES. Using Cliff's delta as a non-parametric effect size measure: An accessible Web App and R tutorial. Practical Assessment, Research, and Evaluation, 2024, 29(1): 2. [doi: 10.7275/pare.1977]
[50] Zhu WM. p<0.05, < 0.01, < 0.001, < 0.0001, < 0.00001, < 0.000001, or < 0.0000001… Journal of Sport and Health Science, 2016, 5(1): 77–79. [doi: 10.1016/j.jshs.2016.01.019]
[51] McHugh ML. The chi-square test of independence. Biochemia Medica, 2013, 23(2): 143–149. [doi: 10.11613/BM.2013.018]
[52] Gavrikov P, Keuper J. Can biases in ImageNet models explain generalization? In: Proc. of the 2024 IEEE/CVF Conf. on Computer Vision and Pattern Recognition. Seattle: IEEE, 2024. 22184–22194. [doi: 10.1109/cvpr52733.2024.02094]
[53] Gastwirth JL. The estimation of the Lorenz curve and Gini index. The Review of Economics and Statistics, 1972, 54(3): 306–316. [doi: 10.2307/1937992]
[54] Jaroenchai N, Wang SW, Stanislawski LV, Shavers E, Jiang Z, Sagan V, Usery EL. Transfer learning with convolutional neural networks for hydrological streamline delineation. Environmental Modelling & Software, 2024, 181: 106165. [doi: 10.1016/j.envsoft.2024.106165]
[55] Khosla A, Zhou TH, Malisiewicz T, Efros AA, Torralba A. Undoing the damage of dataset bias. In: Proc. of the 12th European Conf. on Computer Vision. Florence: Springer, 2012. 158–171. [doi: 10.1007/978-3-642-33718-5_12]
[56] Koch B, Denton E, Hanna A, Foster JG. Reduced, reused and recycled: The life of a dataset in machine learning research. In: Proc. of the 35th Int'l Conf. on Neural Information Processing Systems. 2021. 1.
[57] Crawford K, Paglen T. Excavating AI: The politics of images in machine learning training sets. AI & Society, 2021, 36(4): 1105–1116. [doi: 10.1007/s00146-021-01162-8]
[58] Birhane A, Prabhu VU. Large image datasets: A pyrrhic win for computer vision? In: Proc. of the 2021 IEEE Winter Conf. on Applications of Computer Vision. Waikoloa: IEEE, 2021. 1536–1546. [doi: 10.1109/wacv48630.2021.00158]
[59] Geirhos R, Jacobsen JH, Michaelis C, Zemel R, Brendel W, Bethge M, Wichmann FA. Shortcut learning in deep neural networks. Nature Machine Intelligence, 2020, 2(11): 665–673. [doi: 10.1038/s42256-020-00257-z]
[60] D'Amour A, Heller K, Moldovan D, *et al*. Underspecification presents challenges for credibility in modern machine learning. The Journal of Machine Learning Research, 2022, 23(1): 226.
[61] Cavusoglu H, Li ZL, Huang KW. Can gamification motivate voluntary contributions? The case of StackOverflow Q&A community. In: Proc. of the 18th ACM Conf. Companion on Computer Supported Cooperative Work & Social Computing. Vancouver: ACM, 2015. 171–174. [doi: 10.1145/2685553.2698999]

附中文参考文献

[3] 何润之, 周明辉. 智能时代的软件供应链风险及应对技术. 计算, 2025, 1(3): 59–66. [doi: 10.11991/cccf.202507009]
[13] 纪守领, 王琴应, 陈安莹, 赵彬彬, 叶童, 张旭鸿, 吴敬征, 李昀, 尹建伟, 武延军. 开源软件供应链安全研究综述. 软件学报, 2022, 34(3): 1330–1364. http://www.jos.org.cn/1000-9825/6717.htm [doi: 10.13328/j.cnki.jos.006717]
[14] 高恺, 何昊, 谢冰, 周明辉. 开源软件供应链研究综述. 软件学报, 2024, 35(2): 581–603. http://www.jos.org.cn/1000-9825/6975.htm [doi: 10.13328/j.cnki.jos.006975]
[20] 马俊, 周凯, 任怡, 朱浩, 秦莹, 王静. 面向操作系统版本构建的软件包依赖关系分析. 计算机工程与科学, 2021, 43(11): 1926–1933. [doi: 10.3969/j.issn.1007-130X.2021.11.004]
[21] 王梓博, 贾相堃, 应凌云, 苏璞睿. 面向 PyPI 生态系统的漏洞影响范围细粒度评估方法. 软件学报, 2024, 35(10): 4493–4509. http://www.jos.org.cn/1000-9825/6959.htm [doi: 10.13328/j.cnki.jos.006959]

作者简介

姚思梦, 博士生, 主要研究领域为软件工程理论与方法学.
张洋, 博士, 助理研究员, CCF 高级会员, 主要研究领域为开源软件, 开源生态.
赵佳林, 博士生, 主要研究领域为软件工程理论与方法学.
李俊辰, 硕士生, 主要研究领域为软件工程理论与方法学.
沈阳, 博士生, CCF 学生会员, 主要研究领域为群智范式.
王涛, 博士, 副研究员, CCF 高级会员, 主要研究领域为群智范式.
张迅晖, 博士, 助理研究员, 主要研究领域为智能软件开发.