

开源社区中项目热度阶段预测方法研究——以大语言模型项目为例

卢小艺^{1,2}, 陈阳^{1,3}, 傅晓明^{3,4}, 李聪^{2,3}

- (1. 复旦大学计算与智能创新学院, 上海 200438;
2. 复旦大学未来信息创新学院自适应网络与控制研究室, 上海 200438;
3. 复旦大学社会智能研究中心, 上海 200438;
4. 哥廷根大学计算机科学研究所, 哥廷根 37073)

摘要: 近些年, GitHub 开源社区的累计注册开发者数量已超过1亿, 不同项目在用户关注度上的竞争日益激烈。理解开源社区中项目热度变化不仅有助于研究者和开发者把握技术趋势, 还能为资源配置、合作选择和科研投入提供参考。然而, 现有研究多将热度视为关注度的数值回归或流行度二分类问题, 忽略了热度随时间演化所呈现的阶段性特征及其模糊过渡。为此, 以开源大语言模型项目为研究对象, 构建了热度阶段划分与预测一体化框架, 实现了热度阶段的自动刻画与预测。具体而言, 先在固定窗口内提取项目关注度增长速率与总量构建特征向量, 采用高斯混合模型进行软聚类, 获得4类可解释的热度阶段及其概率标签; 随后, 设计基于时间融合变换器 (temporal fusion transformer, TFT) 的预测模型, 利用包含关注度、分支数、提交数等五维行为指标的序列预测未来窗口内的阶段概率分布。实验结果表明, 该方法在交叉熵、均方误差、平均绝对误差和准确率等指标上优于已有方法, 并在4类热度演化模式中均保持较高精度, 验证了其泛化能力。该方法为开源项目提供了热度阶段预测框架, 具有良好的实用与推广价值。

关键词: 开源社区; 大语言模型; 机器学习

中图分类号: TP181

文献标志码: A

doi: 10.11959/j.issn.2096-6652.202542

Research on popularity stage prediction methods for open-source community projects: a case study of large language model projects

LU Xiaoyi^{1,2}, CHEN Yang^{1,3}, FU Xiaoming^{3,4}, LI Cong^{2,3}

1. College of Computer Science and Artificial Intelligence, Fudan University, Shanghai 200438, China
2. Adaptive Networks and Control Laboratory, College of Future Information Technology, Fudan University, Shanghai 200438, China
3. Research Center for Social Intelligence, Fudan University, Shanghai 200438, China
4. Institute of Computer Science, University of Göttingen, Göttingen 37073, Germany

Abstract: In recent years, the cumulative number of registered developers on the open-source community GitHub has exceeded 100 million, leading to increasingly fierce competition for user attention among projects. Understanding the dynamics of project popularity in open-source communities not only helps researchers and developers grasp technological trends, but also provides valuable references for resource allocation, collaboration decisions, and research investment. However, existing studies often treat popularity as a continuous numerical regression or binary classification task, overlooking the stage-wise evolutionary patterns and gradual transitions exhibited in real project popularity dynamics. Open-

收稿日期: 2025-09-26; 修回日期: 2025-12-02

通信作者: 李聪, cong_li@fudan.edu.cn

基金项目: 国家自然科学基金项目 (No.62173095, No.U23A20331, No.62573133, No.62072115); 上海市科技创新行动计划项目 (No.22510713600)

Foundation Items: The National Natural Science Foundation of China (No.62173095, No.U23A20331, No.62573133, No.62072115), Shanghai Science and Technology Innovation Action Plan Project (No.22510713600)

source large language model (LLM) projects were taken as the research object. An integrated framework for popularity stage segmentation and prediction was proposed, enabling the automatic characterization and forecasting of project popularity stages. Specifically, feature vectors based on growth rate and cumulative scale of project attention within fixed windows were extracted, and a Gaussian mixture model (GMM) for soft clustering was employed to obtain four interpretable stages of popularity with probabilistic labels. Subsequently, a prediction model based on the temporal fusion transformer (TFT) was designed, which leveraged sequential data of five behavioral indicators, such as attention, number of forks, and number of commits, to forecast the stage probability distribution in future windows. Experimental results demonstrate that the proposed method outperforms existing baselines in terms of cross-entropy, mean squared error, mean absolute error, and accuracy, while maintaining high precision across all four patterns of popularity evolution, thereby validating its generalization capability. The proposed approach provides a practical and scalable framework for popularity stage prediction of open-source projects, offering strong utility and potential for applications.

Key words: open-source community, large language model, machine learning

0 引言

近年来,大语言模型 (large language model, LLM) 相关的开源项目数量呈现显著的指数级增长。在 GitHub 平台上利用关键词检索公开项目并按年份统计,发现相关项目数量由 2019 年的约 1 100 个快速增加至 2024 年的约 200 000 个。项目热度已成为社区、平台与研究者共同关注的核心指标。已有研究^[1-2]普遍聚焦于在单一时间点上统计关注度 (star)、提交数 (commit)、派生数 (fork) 等静态信息,并基于这些指标对项目进行热度排序。这些成果为项目热度研究提供了数据和分析范式,也为后续探索热度的时序演化奠定了坚实基础。然而,现有方法主要停留在对这些静态信息的分析,未能结合热度随时间动态演变的过程。

本文聚焦开源社区项目热度的动态变化过程,提出了热度阶段划分与预测方法。具体而言,本文基于代表性在线开源开发者社区 GitHub 可实时获取的多维时间序列数据,综合考虑累计关注度及短期增速,将大语言模型项目的热度演化划分为多个具有语义可解释性的阶段,并在此阶段标签上训练预测模型以输出未来时间窗的阶段概率,从而为项目治理与资源调度提供前瞻性信号。在实验设计中,本文利用 GitHub API 实时获取的五行行为序列数据,包括关注度 (star)、提交数 (commit)、派生数 (fork)、版本发布数 (release) 及问题反馈数 (issue),对近 5 年热门大语言模型相关项目的热度阶段进行了回归式预测。本文方法首先预测阶段的概率分布,并将其中概率最大的阶段作为最终分类标签,以满足项目最可能处于何种热度阶段的

实际需求。实验结果表明,在这一分类判定下,全部样本预测的平均准确率达到 89%,其中早期爆发模式子集的平均准确率进一步提升至 95%,验证了所提方法在热度阶段预测中的泛化性。

本文主要贡献有如下两方面。

(1) 理论贡献——热度演化特征建模:为了识别不同类型项目的热度发展路径,并为后文基于聚合型层次聚类 (Ward 最小方差准则),将大语言模型的项目热度曲线归纳为 4 类热度演化模式 (后期爆发型、线性增长型、热度减缓型、早期爆发型),并针对不同项目定位提出对应的运营与治理建议。

(2) 技术贡献——热度阶段预测框架与实验验证:本文提出基于时间融合变换器 (temporal fusion transformer, TFT)^[3] 的热度阶段预测框架,结果显示,在所构建的数据集上,该模型的热度阶段预测准确率达到 89%。进一步地,在 4 类热度演化模式上分别进行实验,发现模型在不同类型的项目中均能保持较高的准确度,验证了该框架的泛化能力。此外,本文将该框架应用于近期大语言模型相关项目,并对预测结果进行分析,说明该框架可用于识别热度演化趋势并提供前瞻性判断。

第 1 节综述相关工作,回顾聚类方法在开源项目研究中的应用及项目热度研究的静态与动态两条路径,阐明本文选择所用技术方法的原因;第 2 节介绍数据来源、抓取规则与预处理流程;第 3 节对项目关注度曲线进行聚类,识别典型热度演化模式并分析其行为差异;第 4 节提出基于 TFT 的热度阶段预测框架;第 5 节详细描述了实验参数配置,通过不同实验设置下的对比与不同热度变化模式下的验证,全面评估所提方法的性能,并进一步应用于

2025年上半年创建的相关项目，以验证其在真实演化场景中的适用性；第6节对本文贡献进行了系统总结，并展望未来工作。

1 相关工作

本节综述了开源项目热度研究的相关工作。相关研究主要包括3个方面：（1）开源社区项目热度的建模与预测方法，系统回顾已有研究对项目热度建模与预测的探索；（2）聚类方法在开源项目研究中的应用，重点介绍本文采用的聚类方法；（3）多步时间序列预测方法，阐明其代表性进展及本文选择该方法的动机。

1.1 开源社区中项目热度研究

针对开源社区项目热度，目前主流研究大致可分为静态特征分类与动态时间建模两类路径。

一类研究基于关注度等指标的静态数值预测项目是否流行，即热度分类建模。Han等^[1]构建融合GitHub与Stack Overflow特征的分类模型，使用随机森林方法判断项目是否流行，发现分支数、贡献者活跃度等特征具有良好区分性。Fan等^[2]基于人工智能相关项目发现，自述文件中链接数、图片数及是否包含证书是影响项目受欢迎程度的重要因素。这类研究通过变量回归、特征相关性分析等手段揭示了多个静态因素与热度的关系，模型简单可解释。

另一类研究采用时间序列建模，试图捕捉项目热度的动态变化。Sahin等^[4]通过构建循环神经网络（recurrent neural network, RNN）模型，预测开源项目未来关注度的增量趋势，实验证明融合用户影响力（如关注者数量）与行为特征能显著提升预测精度。Borges等^[5]进一步比较了通用回归模型与增长模式特定模型在热度预测上的差异，显示不同增长节奏下模型性能存在显著波动。Borges等^[6]在另一项研究中进一步引入项目排名预测任务，相比于仅预测关注度数值，该方法能够揭示项目在整体平台中的相对受关注程度。Varuna等^[7]以GH Archive数据为基础，基于GitHub多事件构建多变量长短期记忆网络（long short-term memory, LSTM）模型，成功预测了项目、语言、领域等对象的热度走势，验证GitHub活动数据的可预测性。

1.2 聚类方法及其在开源项目研究中的应用

聚类方法因其无须预置标签^[8-9]、能够从复杂数据中自动分辨潜在结构^[10-11]，近年来在开源项目

研究中被广泛采用。Bonfanti^[12]通过对多维时间序列（如提交数、拉取请求数、讨论数等）进行分段与聚类，将局部行为划分为“快速增长”“稳定发展”“停滞平台”等基本模式，再结合层次聚类将整个项目归纳为典型的发展轨迹类型（如集中爆发型、长期稳增型、多周期波动型）。PHANTOM框架^[13]则从实际工程需求出发，通过时间序列特征提取与K-means聚类对GitHub项目进行分类，自动甄别“工程化项目”与“非工程化项目”。Liu等^[14]使用潜变量Hawkes过程模型，对GitHub上的行为事件进行聚类与触发机制建模，发现项目演化中存在不同类型的事件流模式，如“爆发型拉取请求驱动流”与“持续型关注度演进流”等，展示了行为节奏的差异性。此外，Zhou等^[15]在对热门深度学习项目的分析中，使用关注度增长轨迹进行聚类，识别出“爆发式（explosive）”“稳定式（steady）”“后减速式（faster then slower）”3种模式，并结合自述文件结构、代码示例等特征识别了影响热度演化的驱动因素。

综上，现有研究大多依赖硬划分聚类方法，即每个样本只能被唯一归入某一类。然而在项目热度演化过程中，热度阶段多表现为模糊的边界而非完全分离。例如，项目在热度增长长期末端可能已表现出部分爆发期的特征，这使得单一的硬标签难以真实反映阶段演化的连续性。此外，硬划分方法只能给出确定的类别归属，缺乏对分类结果的不确定性描述，而概率解释能够提供样本属于各阶段的概率分布，从而揭示模型对结果的信心程度。因此，本文选择高斯混合模型（Gaussian mixture model, GMM）^[16]进行热度阶段划分。

其基本假设是观测数据由 K 个高斯分量的线性组合生成。模型的概率密度函数定义为：

$$p(x|\theta) = \sum_{k=1}^K \pi_k N(x|\mu_k, C_k) \quad (1)$$

其中，单个分量的多元高斯密度函数为：

$$N(x|\mu_k, C_k) = \frac{1}{(2\pi)^{d/2} |C_k|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T C_k^{-1}(x-\mu_k)\right) \quad (2)$$

其中， $\theta = \{\pi_k, \mu_k, C_k\}_{k=1}^K$ 为参数集合，混合系数 π_k 满足 $\pi_k \geq 0$ ， $\sum_{k=1}^K \pi_k = 1$ ， $\mu_k \in R^d$ 为均值向量， $C_k \in R^{d \times d}$ 为协方差矩阵， $\det(C_k)$ 表示其行列式， d 为特征

维度。

1.3 时间序列预测方法

时间序列预测作为经典的研究问题，已发展出丰富的方法体系^[17]。早期方法多基于统计建模，如自回归移动平均（autoregressive moving average, ARMA）模型^[18]与自回归积分滑动平均（autoregressive integrated moving average, ARIMA）模型^[19]，能够在平稳性假设下捕捉线性相关结构，但难以处理非线性特征与复杂依赖关系。随着机器学习的发展，支持向量回归（support vector regression, SVR）^[20]、随机森林（random forest）^[21]等方法被应用于时间序列任务，在一定程度上提升了对非线性模式的拟合能力。

近年来，深度学习方法在时间序列预测中逐渐占据主流地位。循环神经网络^[22]及其改进结构长短期记忆网络^[23]与门控循环单元（gated recurrent unit, GRU）^[24]能够建模序列中的动态依赖，适用于多种序列场景；卷积神经网络（convolutional neural network, CNN）^[24]也被用于捕捉局部时间模式^[25]；而基于注意力机制的变换器（Transformer）^[26]结构，则进一步提升了对长时依赖的捕获能力^[27]。

在此背景下，时间融合变换器（TFT）^[3]作为近年来提出的代表性模型，在多源输入建模、长短期依赖捕获与预测可解释性等方面表现突出。在结构层面，TFT由以下关键组件构成。

（1）变量选择网络（variable selection network）：对静态、历史、未来3类输入分别学习稀疏权重，实现按变量的自适应筛选与表征对齐。

（2）门控残差网络（gated residual network, GRN）：以门控残差单元稳定深层训练、抑制无关支路，提升对噪声的鲁棒性。

（3）长短期记忆网络编码与解码：捕获局部与短期顺序依赖，作为时序主干。

（4）多头自注意力的时间融合：在解码端对历史与上下文进行全局依赖聚合，提供可解释的时间注意力权重。

（5）量化输出层：以分位数回归直接产生多分位预测，从而给出点预测与不确定性区间。

TFT可以在保持高预测精度的同时，提供可解释的特征贡献与时序依赖刻画，在建模长短期依赖、输出可解释预测结果方面具有明显优势。由于本文的研究目标是预测大语言模型项目在未来时间

窗口内的热度阶段分布，需要同时兼顾预测精度、特征可解释性和对长时依赖的建模能力，因此选择TFT作为核心预测模型是最优方案。

2 数据采集与处理

GitHub是当前全球规模最大、影响力最广的开源代码托管与协作平台，基于Git版本控制系统构建^[28]。其用户数量已超过1亿，累计托管数亿个代码项目，涵盖从个人实验到大型企业级项目的广泛场景^[29]。凭借庞大的用户群体与活跃的协作机制，GitHub已经成为研究开源社区动态与项目热度演化的最具代表性的平台。用户可以通过创建项目存储和管理代码，同时利用平台提供的各类协作功能实现多人协作与社区互动^[30]。

为构建热度阶段预测所需的时间序列数据集，本文以GitHub为数据源，数据抓取规则见表1，筛选项目创建时间从2019年1月1日至2024年12月31日的项目，在此基础上按关注度排序选取前200个项目作为研究对象。随后采用事件级抓取策略：自各项目创建日起，获取其全部关注（star）、提交（commit）、派生（fork）、版本发布（release）与问题事件（issue），所有事件解析后按类型分目录写入文件。接着以项目创建日为起点构建连续日历，将事件时间戳统一到指定时区并映射至自然日，对5类事件按日频数统计（无事件的自然日记为0），最终形成每个项目的日度行为数据表，包含关注度、提交数、派生数、版本发布数、问题反馈数5列，作为后续阶段划分与概率预测模型的输入。

表1 数据抓取规则

项目	规则
数据源	GitHub (REST API)
检索关键词	llm; large language model; llama; glm; gpt; deepseek
时间窗	2019-01-01 至 2024-12-31
主语言限定	Python
排序规则	按关注度降序
样本规模	前200个项目
时区与聚合粒度	统一到指定时区, 按自然日聚合
数据存储	按事件类型分目录存储

此外，为验证模型在近期大语言模型相关项目上的适用性，本文在第5.4节近期大语言模型项目热度阶段预测与分析部分按照完全一致的数据抓取与处理规则，仅调整时间窗口，另行采集了2025年

1月至2025年6月新增的大语言模型相关项目，用于后续预测分析。需要特别说明的是，本文所用数据均来源于GitHub平台公开信息，不涉及任何用户隐私或敏感数据，不存在伦理与合规风险。

3 项目热度变化模式分析

3.1 热度曲线聚类

为刻画开源大语言模型项目在关注度积累过程中的典型演化模式，本文选取2019年1月至2024年12月建立的大语言模型项目关注度作为研究对象，并对其进行标准化与聚类分析，从而为热度阶段预测模型的解释提供更多语义支撑。具体步骤如下。

(1) 序列归一化：读取每个项目的关注度数据，按日累加得到累计关注度曲线，采用线性归一化^[31]将关注度曲线 Star_{cum} 映射至 $[0, 1]$ 区间。

(2) 等距采样与特征表示：在归一化时间轴上均匀采样 N_s 个点，将每个项目的热度曲线统一表示为长度固定的向量 $\mathbf{x}_r \in R^{N_s}$ (r 为项目索引)，避免了不同项目序列长度不一致的问题。

(3) 层次聚类与类别确定：在特征矩阵 $\mathbf{X} \in R^{N_r \times N_s}$ 上 (N_r 表示项目总数)，采用基于 Ward 最小方差^[32]的凝聚型层次聚类 (agglomerative hierarchical clustering) 方法^[33]。记 $\mathbf{x}_j \in R^{N_s}$ 为项目 j 的特征向量。给定一个簇划分 $\{G_1, G_2, \dots, G_T\}$ ，定义簇内平方和 (within-cluster sum of squares, WCSS) 为：

$$\text{WCSS}(T) = \sum_{i=1}^K \sum_{\mathbf{x}_j \in G_i} \left| \mathbf{x}_j - \boldsymbol{\mu}_i \right|^2, \boldsymbol{\mu}_i = \frac{1}{|G_i|} \sum_{\mathbf{x}_j \in G_i} \mathbf{x}_j \quad (3)$$

Ward 方法在每一步合并时，选择使 WCSS 增量最小的两簇进行合并，其增量为：

$$\Delta(G_a, G_b) = \frac{n_a n_b}{n_a + n_b} \left| \boldsymbol{\mu}_a - \boldsymbol{\mu}_b \right|^2 \quad (4)$$

其中， n_a, n_b 为簇大小， $\boldsymbol{\mu}_a, \boldsymbol{\mu}_b$ 为簇均值。随着 T 增加，WCSS 单调减小，但其下降幅度逐渐减弱；当曲线在某一点后减小速度明显放缓时，对应的 T 即为合理的类别数。

(4) 原型曲线构建与可视化：在聚类完成后，对同一类别的热度曲线求均值，获得该类别的平均曲线 (原型曲线) $\overline{\text{Star}}_c$ ， c 代表第 c 类曲线。该曲线用于解释该簇的热度演化特征。

3.2 热度变化模式分析

为确定最优热度演化模式数目，本节绘制了聚类数 T 与 WCSS 的变化曲线，WCSS 随聚类数 T 的

变化如图1所示。其中，虚线表示端点连线，用于度量 WCSS 曲线的弯折位置；浅黄色区域表示簇数不足阶段，WCSS 下降较快；浅灰色区域表示簇数过多阶段，下降幅度趋缓。

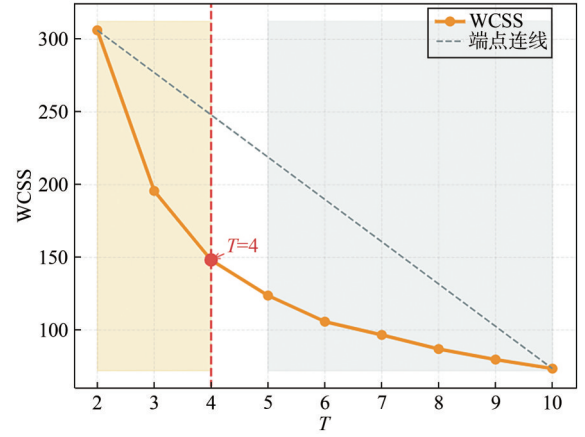


图1 WCSS随聚类数 T 的变化

从图1可以看出，随着聚类数即热度演化模式数目 T 的增加，WCSS 逐渐减小，表明更多的簇划分可以降低类内差异。然而，当 T 增加到4之后，WCSS 的下降幅度明显趋缓，相比之前的改善效果显著减弱，曲线在此位置形成拐点。图1中虚线所示的端点连线 (endpoint line) 用于辅助识别拐点位置，其中曲线到虚线垂直距离最大的点，即 $T=4$ ，被确定为最优聚类数。

高热度项目以及各类项目平均关注度随时间演化曲线如图2所示。其中， $t \in [0, 1]$ 表示归一化时间轴， $\text{Star}_{\text{cum}} \in [0, 1]$ 表示归一化后的关注度曲线， $\overline{\text{Star}}_c$ 表示第 c 类的平均曲线。灰色线为各项目的个体曲线，彩色线为平均曲线。实验设置： $N_s=70$ ， $N_r=200$ 。

为进一步分析各类项目的特征，本节计算了各类演化模式下项目的日均指标，包括每日提交数、每日分支数、每日版本发布数与每日问题反馈数，各热度演化模式下项目日均指标见表2。

在不同热度演化模式下，项目维护者在资源投入与运营策略上的侧重点应有所区别。对各类曲线的行为特征进行归纳不仅能够揭示热度变化的驱动因素，还能为项目开发者与维护团队提供相应治理建议，使他们更好地把握热度波动带来的机遇并应对挑战。

(1) 后期爆发型：图2(a)表明该类项目在早期关注度积累缓慢，至后期出现显著拐点，热度

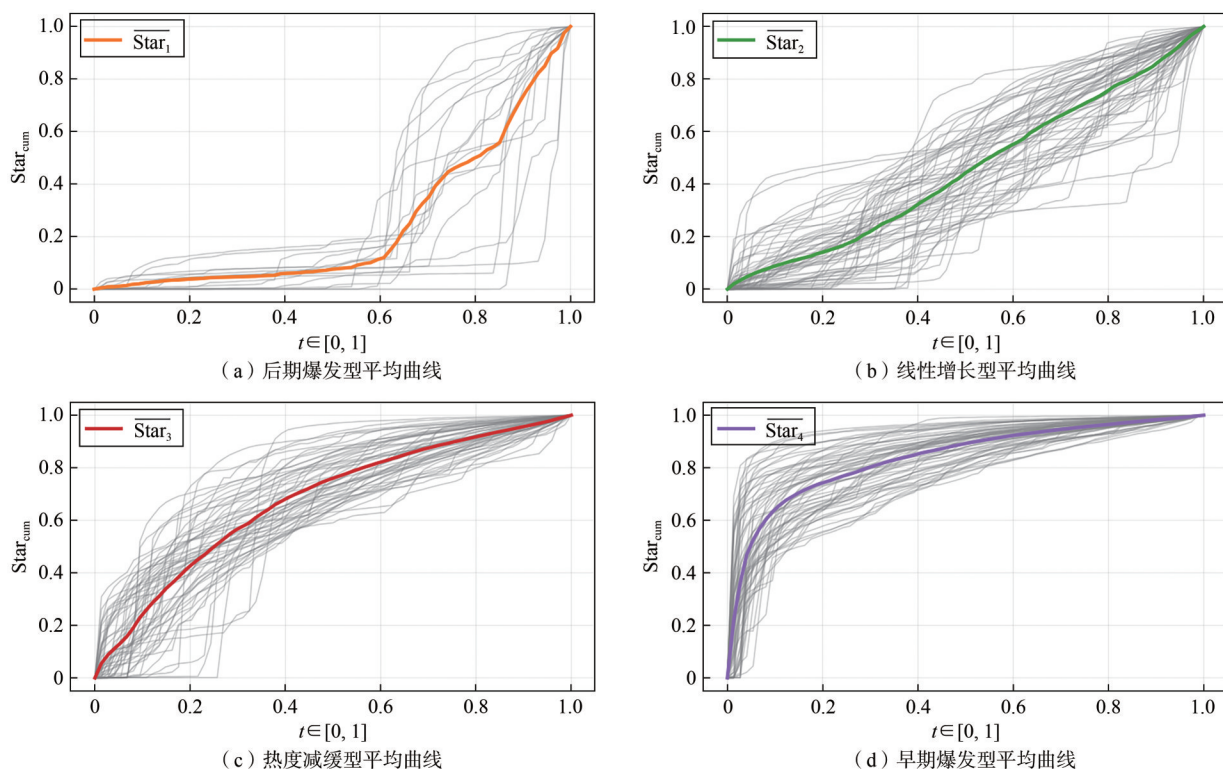


图2 高热度项目以及各类项目平均关注度随时间演化曲线

表2 各热度演化模式下项目日均指标

指标	后期爆发型	线性增长型	热度减缓型	早期爆发型
提交数	2.22	2.86	1.89	1.26
分支数	3.04	1.90	2.79	4.82
版本发布数	0.079	0.087	0.045	0.035
问题反馈数	0.56	1.35	0.69	0.76

迅速飙升。表2显示其分支与版本发布活动频繁，而问题反馈较少，说明生态整合与版本迭代是驱动热度增长的主要因素。因此，维护团队可以考虑在潜在爆发前提前进行宣传与插件准备，而在爆发后将重点转向接口稳定性与生态维护。

(2) 线性增长型：图2(b)表明该类关注度曲线以近似恒定的斜率上升，无明显爆发节点。表2显示该类项目在提交数、版本发布数和问题反馈数上均表现活跃，显示持续迭代和社区互动共同支撑了热度稳定增长。在这种情形下，维护团队可以采取保持长期资源投入的策略，以维持线性增长的可持续性。

(3) 热度减缓型：图2(c)表明该类项目在初期迅速升温，但中后期增速显著放缓。表2显示该类项目尽管分支活动频繁，但官方版本更新有限，表明社区驱动的衍生开发多于官方投入。典型

代表是Meta公司开源的大语言模型项目meta-llama/llama，模型发布初期被广泛引用，但后续更新趋缓。对于此类项目，维护团队可以考虑加强文档整理与迁移指南，并通过适度更新来维持项目曝光度。

(4) 早期爆发型：图2(d)表明此类项目在发布初期关注度迅速攀升，随后进入平台期。表2显示其分支数最高，反映模型权重一经公开即引发集中关注，但代码更新频率较低。对于这类项目，维护团队可以采取上线前准备充足的下载镜像和多语言权重的方式，以应对高并发访问，并在热度趋稳后将工作重心转向版本迭代或插件开发。

4 大语言模型项目热度阶段预测方法设计

为实现对开源大语言模型项目热度阶段的系统建模与预测，本文提出TFT-Stage框架。该框架由以下环节组成。

(1) 热度阶段构建：基于项目关注度增长速率与关注度总规模，利用无监督聚类方法对时间片段进行划分，从而得到热度阶段标签。该构建方法能够根据项目的实际关注度状态定义其相对热度阶段，确保阶段标签具备明确的语义性与可解释性。

(2) 预测模型构建：在得到阶段标签后，将未来阶段分布预测建模为时间序列的多输出概率回归问题。本文采用深度学习模型TFT^[3]，充分考虑时间依赖性与多特征交互，预测未来时间窗口内各阶段的概率分布。

最后，对后续实验使用的评价指标进行说明，指标主要包括交叉熵、均方误差、平均绝对误差以及准确率。这些指标为框架预测性能的定量分析提供了方法支撑。

4.1 热度阶段构建

针对大语言模型开源项目热度阶段的自动构建，本文采用两个特征，增长速率 g_t 与规模对数 c_t ，分别刻画项目关注度的相对增势与绝对体量。

(1) 增长速率 g_t ：衡量当前时间片段内的关注度相对增幅。设窗口 t 内新增关注度为 ΔS_t ，窗口结束时的累计关注度为 S_t ，引入微小常数 ε 避免分母为零，则：

$$g_t = \frac{\Delta S_t}{S_t + \varepsilon} \quad (5)$$

该比值直接反映本期增长占当前体量的比例，可跨项目比较增长动量。

(2) 规模对数 c_t ：用于描述项目累计关注规模，采用对数变换压缩长尾分布：

$$c_t = \log(S_t + 1) \quad (6)$$

对数形式既保留了热度信息，又减轻了大体量项目对模型训练的主导效应。

对每个时间窗口分别计算增长速率 g_t 与规模对数 c_t ，并以此为项目片段的二维特征表示，接着在统一的特征空间上对所有片段进行无监督聚类，从而划分热度阶段。具体步骤如下。

(1) 特征处理：为减少极端值对模型的干扰，本文对关注度增长速率执行符号保持的平方根压缩，并对所有特征执行Z-score标准化^[31]处理，确保不同特征在同一量纲下进行聚类，从而保证聚类结果的可比性与稳定性。

(2) 聚类方法：在特征处理完成后，本文采用高斯混合模型对所有时间片段对应的二维特征向量进行聚类。聚类数，即热度阶段数 K 的选择基于贝叶斯信息准则（Bayesian information criterion, BIC）^[34]。BIC同时考虑拟合优度与模型复杂度，其形式为：

$$\text{BIC} = -2 \ln \hat{L} + k \ln m \quad (7)$$

其中， $\hat{L}$ 为在最大似然估计下的模型对数似然， k 为待估参数个数， m 为样本量。该准则通过 $-2 \ln \hat{L}$ 衡量拟合效果，用 $k \ln m$ 惩罚过多参数，数值越小表示综合表现越优。本文在候选 K 上计算BIC，并选择使BIC最小的 K 作为最终聚类数。

(3) 阶段命名与解释：在得到 K 个聚类后，计算各类在两个特征维度上的均值，便于分析与理解不同热度阶段的特征，并结合分布特征进行解释与命名，作为接下来预测模型的阶段标签。

4.2 热度阶段预测模型构建

本文将热度阶段分布的预测问题建模为多输出概率回归。具体步骤如下。

(1) 数据集构建

对每个项目，在时刻 t 结束的滑动窗口样本记为：

$$\mathbf{X}^{(t)} = [\mathbf{x}_1^{(t)}, \mathbf{x}_2^{(t)}, \dots, \mathbf{x}_W^{(t)}] \in R^{W \times d} \quad (8)$$

其中， $\mathbf{x}_s^{(t)} \in R^d$ 为窗口内第 s 天($s=1, 2, \dots, W$)的行为特征向量，包含“关注度、提交数、分支数、版本发布数、问题反馈数”5个维度。设未来预测区间长度为 h 。对区间内的每一日 $t+\tau$ ($\tau=1, 2, \dots, h$)，依据GMM聚类的阶段划分结果，得到一条软标签向量 $\mathbf{r}_{t+\tau}$ ，其为 K 维概率向量（各分量非负且和为1）， K 为热度阶段数。将该区间内的软标签按日平均，得到用于监督的阶段概率向量：

$$\mathbf{q}^{(t)} = \frac{1}{h} \sum_{\tau=1}^h \mathbf{r}_{t+\tau} \quad (9)$$

其中， $\mathbf{q}^{(t)}$ 同为 K 维概率向量。最终， $(\mathbf{X}^{(t)}, \mathbf{q}^{(t)})$ 作为一对样本-标签，用于后续模型训练。

(2) 特征建模与表示学习

本文在特征建模部分采用TFT作为核心模型。TFT首先通过变量选择网络对输入的多维行为特征分配可学习权重，从而自动突出关键变量、抑制冗余特征。随后，经由GRN对输入进行非线性变换并保持训练过程的稳定性。对于时间依赖的建模，编码端使用LSTM捕获局部和短期的顺序特征；在解码端，TFT引入多头自注意力机制，对历史序列与上下文信息进行全局融合，从而兼顾短期依赖与长期趋势。

(3) 概率预测与训练优化

设第 i 个样本的真实阶段概率分布为：

$$\mathbf{p}_i = (p_{i,1}, p_{i,2}, \dots, p_{i,K}), p_{i,k} \in [0, 1], \sum_{k=1}^K p_{i,k} = 1 \quad (10)$$

其中, p_{ik} 表示样本 i 属于阶段 k 的真实概率。模型预测结果为第 i 个样本在各阶段的概率分布:

$$\hat{\mathbf{p}}_i = (\hat{p}_{i,1}, \hat{p}_{i,2}, \dots, \hat{p}_{i,K}), \hat{p}_{i,k} \in [0, 1], \sum_{k=1}^K \hat{p}_{i,k} = 1 \quad (11)$$

预测分布由各阶段的原始得分 $\hat{y}_{i,k}$ (即第 i 个样本在阶段 k 上的未归一化输出) 经 softmax 变换得到:

$$\hat{\mathbf{p}}_i = \text{softmax}(\hat{y}_{i,1}, \hat{y}_{i,2}, \dots, \hat{y}_{i,K}) \quad (12)$$

训练时采用分位数回归损失, 在分位数 $q=0.5$ 下等价于中位数回归:

$$L = \sum_i \sum_{k=1}^K \rho_{0.5}(p_{i,k} - \hat{y}_{i,k})$$

$$\rho_{0.5}(u) = \max\{0.5u, (0.5-1)u\} \quad (13)$$

使用分位数回归损失可减弱极端增长样本对梯度方向的影响, 并以更平滑的方式刻画阶段概率的连续变化, 同时为未来通过多分位点扩展不确定性区间提供了可能。模型的学习率、批次大小和隐藏层维度等超参数均通过交叉验证选取。

4.3 模型评估

为了对提出的 TFT-Stage 框架在热度阶段预测任务中的性能进行定量评估, 本文从 3 个维度选取了 4 个评价指标^[35]——概率分布相似度 (交叉熵)、数值拟合精度 (均方误差与平均绝对误差) 以及分类一致性 (准确率), 从概率分布相似度与数值误差两个维度刻画预测质量, 并便于后续研究在统一指标体系下进行可比性验证。

(1) 交叉熵 (cross-entropy, CE), 用于度量真实分布与预测分布之间的相似性, 值越小表示预测越接近真实分布:

$$CE = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K p_{i,k} \log \hat{p}_{i,k} \quad (14)$$

(2) 均方误差 (mean squared error, MSE), 对较大偏差施加二次惩罚, 更敏感地反映预测失真:

$$MSE = \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K (p_{i,k} - \hat{p}_{i,k})^2 \quad (15)$$

(3) 平均绝对误差 (mean absolute error, MAE), 相比 MSE 对离群误差更具鲁棒性, 从整体上衡量拟合精度:

$$MAE = \frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K |p_{i,k} - \hat{p}_{i,k}| \quad (16)$$

(4) 准确率 (accuracy, ACC):

$$ACC = \frac{1}{N} \sum_{i=1}^N I\left(\arg \max_k p_{i,k} = \arg \max_k \hat{p}_{i,k}\right) \quad (17)$$

其中, $I(\cdot)$ 为指示函数。该指标将概率分布硬化为单一阶段标签, 统计预测命中比例, 用于反映模型在分类判定下的性能。

5 实验结果

5.1 热度阶段聚类结果

本文选取建立于 2019 年 1 月至 2024 年 12 月的大语言模型项目, 并将其关注度序列划分成长度为 20 天的窗口, 最终确定热度阶段数 $K=4$, 对应 4 种热度阶段: 稳定增长期、爆发期、缓慢增长期与成熟期。

热度阶段划分结果见表 3, 给出了各阶段的平均增长速率与平均关注度规模, 可见 4 类阶段在增长势与体量上具有明显差异: 爆发期具有最高的增长速率, 成熟期具有最高的关注度规模, 而缓慢增长期和稳定增长期处于二者之间, 显示出不同的发展特征。

表 3 热度阶段划分结果

阶段序号	阶段名称	平均增长速率	平均关注度规模
0	稳定增长期	0.137	6.479
1	爆发期	0.682	6.357
2	缓慢增长期	0.043	8.665
3	成熟期	0.011	8.911

为进一步验证聚类阶段划分的合理性, 本文选取清华大学开源的大规模语言模型项目 THUDM/GLM-130B 作为案例。ChatGLM 模型于 2022 年发布, 它在多语言 (尤其是中英文) 理解与生成任务中表现出色, 兼具开源可用性与实际落地能力, 在参数规模、推理速度和对中文生态的优化等方面均有突出优势。一经开源便在社区获得广泛关注, 短时间内积累了大量用户与二次开发者, 成为大语言模型项目中具有代表性的案例。THUDM/GLM-130B 实际关注度与热度阶段随时间变化如图 3 所示。其中, 阶段 0 代表稳定增长期, 阶段 1 代表爆发期, 阶段 2 代表缓慢增长期, 阶段 3 代表成熟期。图 3 (a) 为同一时间内项目关注度曲线, 其中可见: 当项目阶段从增长期跃迁至爆发期时, 累计关注度出现显著陡升, 而进入成熟期后增长趋势趋缓, 热度逐渐稳定。图 3 (b) 展示了该项目自发布以来的热度

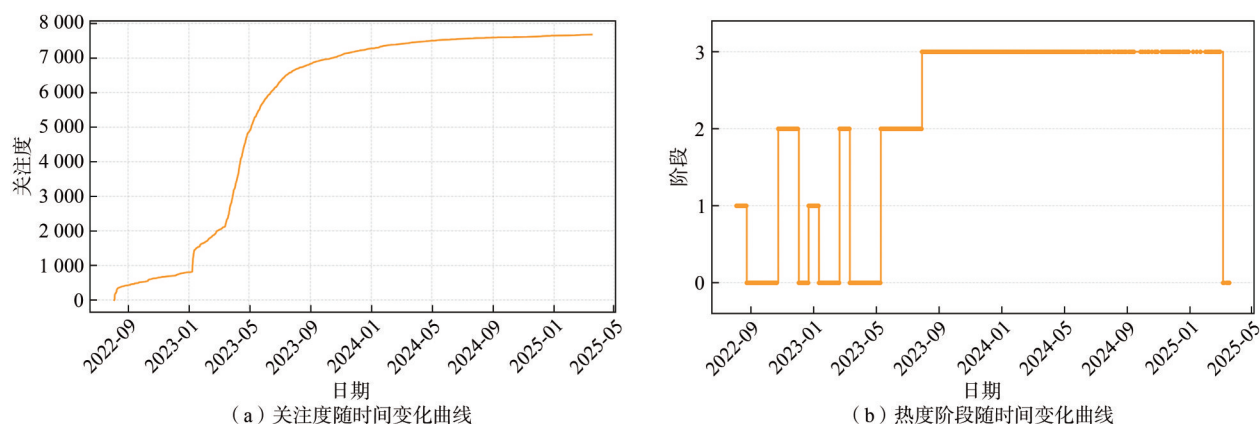


图3 THUDM/GLM-130B实际关注度与热度阶段随时间变化

阶段变化曲线，横轴为日期，纵轴为聚类得到的热度阶段。图3(b)显示：项目早期热度相对平稳，随后多次经历从增长期到爆发期的跃迁，最终进入成熟期，符合典型开源项目的热度演化特征。整体上，关注度变化与热度阶段变化高度一致，验证了本文提出的热度阶段聚类策略能够有效刻画项目热度真实演化过程的结论。

5.2 不同实验设置下的预测性能

为验证 TFT-Stage 框架的有效性，本文系统评估了其在不同输入天数（即不同长度的历史序列）下的预测性能，并与3种主流基线模型 XGBoost^[36]、DLinear^[37]和 TiDE^[38]进行对比。

模型在不同输入天数下的预测性能如图4所示。其中，实验设置 learning rate= 1×10^{-3} , hidden size=64, attention head=4, dropout=0.1, batch size=256, 以项目为单位进行 8:2 训练/验证划分。TFT-Stage 在大多数设置下均取得最优表现，且在输入窗口为 40 天时达到最佳性能 (CE=0.997、MAE=0.240、MSE=0.079、ACC=89%)；XGBoost 虽在短输入条件下表现相对稳健，但整体准确率明显低于 TFT-Stage；DLinear 和 TiDE 在部分场景中取得较低的 MAE，但在 CE 与 ACC 指标上均显著落后。由于实际应用更关注模型对最可能热度阶段的判定，TFT-Stage 在保持较低 CE 的同时实现了更高的准确率，更契合热度预测的业务需求。此外，随着输入窗口的增加，TFT-Stage 的预测性能持续提升并趋于稳定，表明较长的历史上下文能够有效增强热度阶段的判定能力。

5.3 不同热度演化模式下的预测泛化性分析

为了检验 TFT-Stage 框架在不同热度演化模式下的泛化性，本文分别对后期爆发型、线性增长

型、热度减缓型、早期爆发型 4 类热度演化模式进行了热度阶段预测实验，不同热度演化模式下热度阶段预测性能对比见表4。

由表4可见，早期爆发型在4项指标上均表现最好，显示模型能够利用窗口首段的高速增势特征快速识别该阶段；后期爆发型则因爆发信号尚未在窗口内完全显露，准确率相对较低，对其概率分布的刻画也最困难。线性增长型和热度减缓型的性能介于二者之间，且 MAE、MSE 接近整体水平，说明模型对增速平稳或略有放缓的阶段具备合理的拟合能力。

这一结果表明 TFT-Stage 热度阶段预测框架不仅能在整体任务上保持较高准确率，也能在多样化的热度演化模式下稳定发挥作用，从而展现出其在开源项目热度分析中的应用价值与推广潜力。

5.4 近期大语言模型项目热度阶段预测与分析

本文将该方法应用于 2025 年 1 月至 2025 年 6 月新增的大语言模型相关项目中关注度排名前 200 的项目。以 7 月 23 日至 8 月 31 日（40 天）的关注度增长序列为输入，预测其在 9 月 1 日至 9 月 20 日的热度阶段，并据此分析不同热度阶段的分布特征。近期大语言模型相关项目处于各热度阶段的占比见表5。

从整体上看，近期项目主要集中在爆发期和稳定增长期，而成熟期与缓慢增长期的项目相对较少。

为了分析处于爆发期项目的关注方向，本文获取了每个项目在 GitHub 上公开的标签 (topics) 及简介 (description)，并根据其中呈现的主要功能特征对项目进行了人工归类。具体做法为，逐一阅读项目的标签和简介，判断其主要面向的应用场景或

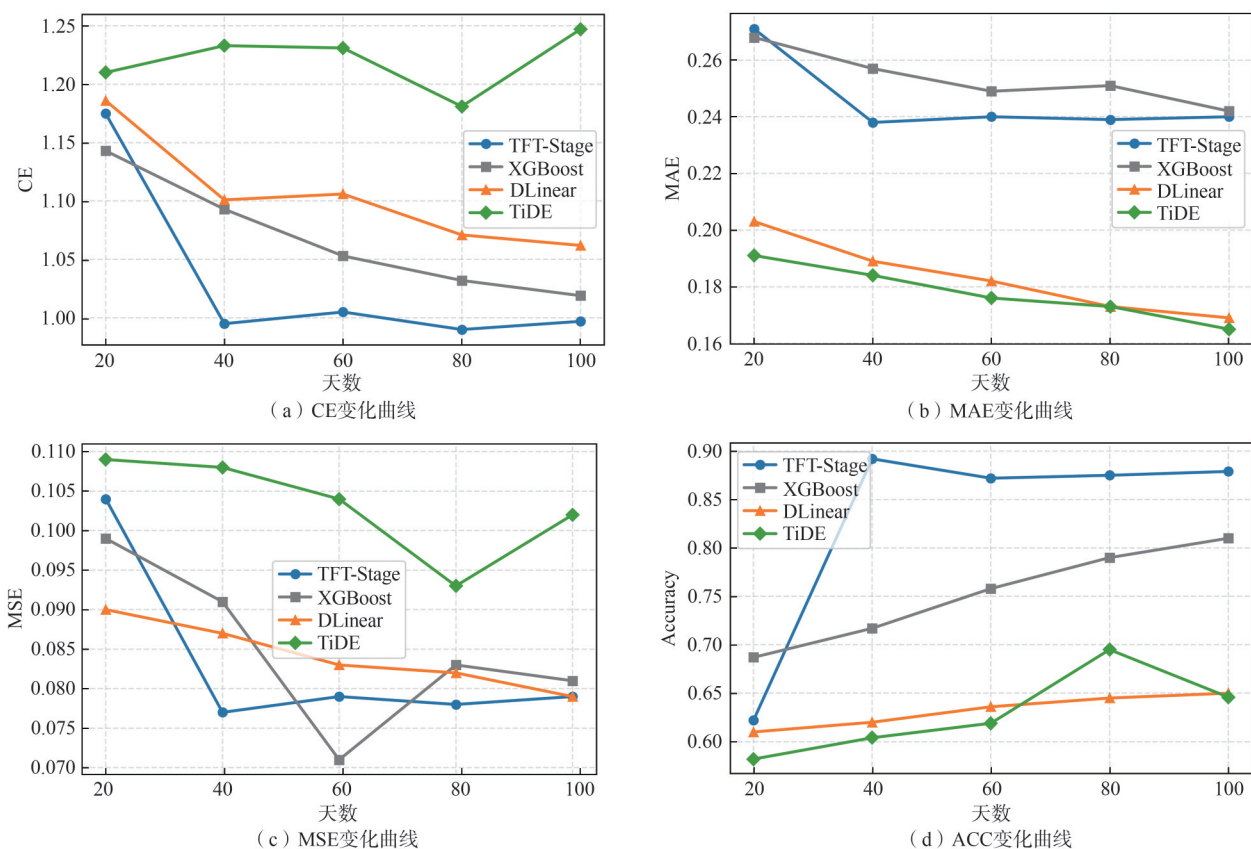


图4 模型在不同输入天数下的预测性能

技术侧重点, 将其归入多智能体协同、自动化推理链、本地部署优化等主题类别, 以概括爆发期项目在技术方向上的整体分布情况。

结果显示, 爆发期项目高度集中于多智能体协同和自动化推理链生成两个方向。其中, 多智能体协同类项目占比达到约47%, 代表性项目有Hazy-Research/minions通过轻量级智能体编排实现复杂任务的模块化协作; 自动化推理链生成类项目占比约34%, 例如, SakanaAI/treequest以树状结构显式组织推理链, OpenManus/OpenManus-RL通过强化学习优化推理质量; 另有14%的项目集中在本地部署与推理加速, 如NVlabs/Fast-dLLM, 在不重新训练的情况下大幅提升生成速度, 并保持输出质量。

表4 不同热度演化模式下热度阶段预测性能对比

类别	后期爆发型	线性增长型	热度减缓型	早期爆发型	整体
CE	1.22	1.02	0.96	0.93	1.00
MAE	0.23	0.23	0.24	0.24	0.24
MSE	0.07	0.07	0.08	0.08	0.08
ACC	0.67	0.82	0.90	0.95	0.89

表5 近期大语言模型相关项目处于各热度阶段的占比

阶段名称	占比
稳定增长期	49.5%
爆发期	33.5%
缓慢增长期	9.0%
成熟期	8.0%

总体而言, 处于爆发期的项目反映出开源社区在该时间窗口内对多智能体、推理链与轻量化部署的高度关注。

6 结束语

本文面向开源社区中项目的演化过程, 提出了一套从热度阶段划分到阶段预测的完整框架, 并采用大语言模型项目进行验证分析。通过高斯混合模型(GMM)对项目在不同时期的关注度增长速率与体量规模进行软聚类, 获得4类可解释的热度阶段及其概率标签, 为后续建模提供结构化监督。在预测建模部分, 本文将阶段预测问题形式化为多输出概率回归任务, 并基于时间融合变换器(TFT)构建回归模型。实验结果显示, TFT在多个评价指

标上均显著优于对比方法,尤其在40天输入窗口下获得最佳效果。该框架具有通用性,能够推广至不同类型的开源项目,从而帮助研究者更系统地识别项目热度阶段。

针对大语言模型项目,本文总结分析其热度变化模式,并提出相应维护与治理建议。实验结果显示,本文提出的方法能够有效刻画大语言模型项目的热度阶段变化,在阶段预测任务中达到了89%的准确率。该框架在大语言模型项目上的应用可为开发团队在算力调度、版本发布节奏以及社区资源投入等环节提供依据,从而提升项目运维与推广的针对性。进一步地,本文将该框架应用于2025年新增的大语言模型相关项目,利用其短期增长序列对后续阶段进行预测,结果显示2025年上半年新创建的项目主要集中于爆发期与稳定增长期,并在爆发期项目中呈现出多智能体协同、自动化推理链生成以及本地部署优化等较为突出的功能方向。该结果表明,所提方法能够在新项目场景中保持稳定的阶段识别能力,并可用于理解新生态中关注点的聚集趋势。

尽管本文提出的热度阶段建模与预测框架在结构设计与实验性能方面表现出较强的实用性与可解释性,但仍存在若干需要进一步完善之处。目前的阶段划分依赖GMM,虽具备软分类的灵活性,但仍受特征空间构造与样本分布影响,可能在边界判定上存在语义模糊问题。此外,输入特征主要集中于动态行为频次,尚未引入项目分类等静态语义信息,可能限制了模型对复杂演化模式的识别能力。未来工作可进一步扩展输入特征维度,并探索模型在实际平台部署的可行性,以增强模型的实用性与推广价值。

参考文献:

- [1] HAN J X, DENG S G, XIA X, et al. Characterization and prediction of popular projects on GitHub[C]//Proceedings of the 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC). Piscataway: IEEE Press, 2019: 21-26.
- [2] FAN Y R, XIA X, LO D, et al. What makes a popular academic AI repository?[J]. Empirical Software Engineering, 2021, 26(1): 2.
- [3] LIM B, ARİK S Ö, LOEFF N, et al. Temporal fusion transformers for interpretable multi-horizon time series forecasting[J]. International Journal of Forecasting, 2021, 37(4): 1748-1764.
- [4] SAHIN S E, KARPAT K, TOSUN A. Predicting popularity of open source projects using recurrent neural networks[C]//Proceedings of the 2019 Open Source Systems. Cham: Springer, 2019: 80-90.
- [5] BORGES H, HORA A, VALENTE M T. Understanding the factors that impact the popularity of GitHub repositories[C]//Proceedings of the 2016 IEEE International Conference on Software Maintenance and Evolution (ICSME). Piscataway: IEEE Press, 2016: 334-344.
- [6] BORGES H, HORA A, VALENTE M T. Predicting the popularity of GitHub repositories[C]//Proceedings of the 12th International Conference on Predictive Models and Data Analytics in Software Engineering. New York: ACM Press, 2016: 1-10.
- [7] VARUNA T V, MOHAN A. Trend prediction of GitHub using time series analysis[C]//Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). Piscataway: IEEE Press, 2019: 1-7.
- [8] 何玉林, 杨锦, 黄哲, 等. 基于标签迭代的聚类集成算法[J]. 智能科学与技术学报, 2024, 6(4): 466-479.
- HE Y L, YANG J, HUANG Z X, et al. Label iteration-based clustering ensemble algorithm[J]. Chinese Journal of Intelligent Science and Technology, 2024, 6(4): 466-479.
- [9] 胡毅, 瞿博阳, 梁静, 等. 进化集成学习算法综述[J]. 智能科学与技术学报, 2021, 3(1): 18-33.
- HU Y, QU B Y, LIANG J, et al. A survey on evolutionary ensemble learning algorithm[J]. Chinese Journal of Intelligent Science and Technology, 2021, 3(1): 18-33.
- [10] 丁宸聪. 基于选择性深度嵌入聚类的复杂未知雷达信号分选方法[J]. 智能科学与技术学报, 2025, 7(3): 329-337.
- DING C C. A complex unknown radar signal deinterleaving method based on selective deep embedding clustering[J]. Chinese Journal of Intelligent Science and Technology, 2025, 7(3): 329-337.
- [11] LI C, SHI M, QU B, et al. Deep attributed network representation learning via attribute enhanced neighborhood[J]. Neurocomputing, 2022, 508: 170-181.
- [12] BONFANTI M. A framework for identifying evolution patterns of open-source software projects[D]. Delft: Delft University of Technology, 2021.
- [13] PICKERILL P, JUNG H J, OCHODEK M, et al. PHANTOM: curating GitHub for engineered software projects using time-series clustering[J]. Empirical Software Engineering, 2020, 25(4): 2897-2929.
- [14] LIU S Z, YAO S C, LIU D X, et al. A latent Hawkes process model for event clustering and temporal dynamics learning with applications in GitHub[C]//Proceedings of the 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). Piscataway: IEEE Press, 2019: 1275-1285.
- [15] ZHOU Y R, QIAO Y, XU T. Exploring the characteristics of popular deep learning GitHub repositories[C]//Proceedings of the 2023 International Conference on Intelligent Computing and Next Generation Networks (ICNGN). Piscataway: IEEE Press, 2023: 1-6.
- [16] CHEN X, ZHANG A Y. Achieving optimal clustering in Gaussian mixture models with anisotropic covariance structures[C]//Proceedings of the 38th International Conference on Neural Information Processing Systems (NeurIPS). New York: Curran Associates, 2024: 113698-113741.
- [17] ZHOU Z L, LI C, VAN MIEGHEM P, et al. Predicting higher-order dynamics with unknown hypergraph topology[J]. IEEE Transactions on Circuits and Systems I: Regular Papers, 2025, 72(4): 1693-1706.
- [18] GÜNEYİ E T, YALDIZ B, CANBOLAT A, et al. Learning graph ARMA processes from time-vertex spectra[J]. IEEE Transactions on Signal Processing, 2024, 72: 47-56.
- [19] HYNDMAN R J, ATHANASOPOULOS G. Forecasting: principles and practice[M]. Melbourne: OTexts, 2018.
- [20] WANG B, YANG A. A consolidated cross-validation algorithm for support vector machines via data reduction[C]//Proceedings of the 36th In-

- ternational Conference on Neural Information Processing Systems (NeurIPS). New York: Curran Associates, 2022, 35: 394-405.
- [21] RHODES J S, CUTLER A, MOON K R. Geometry- and accuracy-preserving random forest proximities[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, 45(9): 10947-10959.
- [22] YAN S, LIANG Y, ZHENG L, et al. Explainable gated Bayesian recurrent neural network for non-Markov state estimation[J]. IEEE Transactions on Signal Processing, 2024, 72: 4302-4317.
- [23] HOCHREITER S, SCHMIDHUBER J. Long short-term memory[J]. Neural Computation, 1997, 9(8): 1735-1780.
- [24] BENIDIS K, RANGAPURAM S S, FLUNKERT V, et al. Deep learning for time series forecasting: tutorial and literature survey[J]. ACM Computing Surveys, 2022, 55(6): 1-36.
- [25] WANG W, LI C, QU B, et al. Predicting epidemic threshold in complex networks by graph neural network[J]. Chaos, 2024, 34(6): 063129.
- [26] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS). New York: Curran Associates, 2017: 5998-6008.
- [27] 姚云, 胡振斌, 邓涛, 等. 基于自适应池化注意力Transformer的唇语识别方法[J]. 智能科学与技术学报, 2025, 7(2): 211-220.
- YAO Y, HU Z X, DENG T, et al. A lip reading method based on adaptive pooling attention Transformer[J]. Chinese Journal of Intelligent Science and Technology, 2025, 7(2): 211-220.
- [28] 杨波, 于茜, 张伟, 等. GitHub开源软件开发过程中影响因素的相关性分析[J]. 软件学报, 2017, 28(6): 1330-1342.
- YANG B, YU Q, ZHANG W, et al. Influence factors correlation analysis in git hub open source software development process[J]. Journal of Software, 2017, 28(6): 1330-1342.
- [29] 王明宇, 宫庆媛, 瞿晶晶, 等. 基于机器学习的GitHub企业影响力分析与预测[J]. 智能科学与技术学报, 2023, 5(3): 330-342.
- WANG M Y, GONG Q Y, QU J J, et al. Analysis and prediction of GitHub company influence based on machine learning[J]. Chinese Journal of Intelligent Science and Technology, 2023, 5(3): 330-342.
- [30] SÁNCHEZ R J M, CIRIA G G, OLIVERO M Á, et al. Feature-based characterization of GitHub OSS projects for performance assessment[C]//Proceedings of the 29th ACM International Systems and Software Product Line Conference-Volume A. New York: ACM Press, 2025: 94-104.
- [31] HAN J W, KAMBER M, PEI J. Data mining: concepts and techniques[M]. 3rd ed. Amsterdam: Elsevier, 2012.
- [32] MURTAGH F, LEGENDRE P. Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion?[J]. Journal of Classification, 2014, 31(3): 274-295.
- [33] KAUFMAN L, ROUSSEEUW P J. Finding groups in data: an introduction to cluster analysis[M]. New York: John Wiley & Sons, 1990.
- [34] GIUDICE E, KUIPERS J, MOFFA G. A Bayesian take on Gaussian process networks[C]//Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS). New York: Curran Associates, 2023: 56602-56614.
- [35] BISHOP C M. Pattern recognition and machine learning[M]. New York: Springer, 2006.
- [36] CHEN T Q, GUESTRIN C. XGBoost: a scalable tree boosting system[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2016: 785-794.
- [37] ZENG A L, CHEN M X, ZHANG L, et al. Are transformers effective for time series forecasting?[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto: AAAI Press, 2023, 37(9): 11121-11128.
- [38] DAS A, KONG W, LEACH A, et al. Long-term forecasting with TiDE: time-series dense encoder[J]. Transactions on Machine Learning Research, 2023: 1-27.

[作者简介]



卢小艺 (2003-), 女, 复旦大学计算与智能创新学院硕士生, 主要研究方向为复杂网络理论及应用、人工智能驱动的网络科学研究。



陈阳 (1981-), 男, 博士, 复旦大学计算与智能创新学院教授、博士生导师, 上海市智能信息处理重点实验室副主任, 主要研究方向为社会计算、计算机网络、大语言模型。



傅晓明 (1973-), 男, 博士, 哥廷根大学计算机科学研究所教授, 复旦大学社会智能研究中心首席科学家, 德国国家工程院院士, 欧洲科学院院士, IEEE Fellow、ACM杰出科学家, 主要研究方向为社会计算、计算机网络等。



李聪 (1986-), 女, 博士, 复旦大学未来信息创新学院副研究员, 主要研究方向为复杂网络理论及应用、人类行为分析、人工智能驱动的网络科学。