基于双层机器学习的业务流程剩余时间预测

孙笑笑 侯文杰 应钰柯 俞东进
(杭州电子科技大学计算机学院 杭州 310018)
(复杂系统建模与仿真教育部重点实验室 杭州 310018)

摘 要 近年来，流程挖掘技术不再局限于对事件日志的线下分析以实现对流程模型的改进，而更加关注如何为业务流程的优化提供在线支持. 其中业务流程剩余执行时间的预测监控是流程挖掘中的关键研究问题，它能为相关者提供及时的预测信息，进而采取有效措施以减少流程执行风险（例如超过时间限制）. 当前剩余时间预测的研究仅考虑单个流程实例的内部属性，而忽略了多个实例共同执行对剩余执行时间所产生的竞争影响. 为此，本文考虑多实例间的资源竞争，并将其作为预测的主要输入属性之一. 此外，本文还通过分析历史事件日志选择出一些对当前流程实例执行时间有重大影响的关键活动，并将其也作为预测的输入属性之一. 同时，为提高预测模型的精度和在复杂应用场景中的适应性，本文利用堆叠技术将 XGBoost 模型和 LightGBM 模型进行融合，构建出双层混合预测模型来完成对业务流程剩余时间的预测. 在四个真实数据集上的实验表明，考虑了实例间属性以及关键活动属性的混合预测模型在平均绝对误差上比 LSTM 和 XGBoost 方法分别降低了 11.6%和 15.8%.

关键词 流程挖掘；流程剩余时间预测；机器学习；资源竞争；融合模型
中图法分类号 TP391 **DOI 号** 10.11897/SP.J.1016.2021.02283

Business Process Remaining Time Prediction Based on Two-Layer Machine Learning

SUN Xiao-Xiao HOU Wen-Jie YING Yu-Ke YU Dong-Jin
(*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou* 310018)
(*Key Laboratory of Complex Systems Modeling and Simulation, Ministry of Education, Hangzhou* 310018)

Abstract In recent years, process mining technology is no longer limited to offline analysis of event logs to improve process models, but concerns more on how to provide online support for business process (BP) optimization. Particularly, predictive process monitoring (PPM) aims at providing predictions of future status of on-going instances such as remaining execution time, the next activity to be executed, execution outcome and resource execution. Accurate predictions of these status help managers to take proactive actions to reduce process execution risks. Among all tasks of PPM, remaining time prediction is the most important one as it provides timely forecast information to help relevant personnel to adjust the priority of activities, thus avoiding the execution of some instances exceeding their deadlines. Current researches on remaining time prediction, however, mostly consider the impact of internal attributes of a single process instance while ignoring the competitive impact of multiple instances executing together. Moreover, the input of most predictions is directly extracted from event logs without in-depth analysis. Therefore, in this paper, resource competition is firstly defined by several inter-instance attributes as the main input of the

收稿日期：2020-02-11；在线发布日期：2020-08-19. 本课题得到国家自然科学基金（61472112）、浙江省自然科学基金（LQ20F020017）、浙江省重点研发资助项目（2020C01165，2017C01010）资助. **孙笑笑**，博士，讲师，中国计算机学会（CCF）会员，主要研究领域为业务流程管理、时空数据挖掘. E-mail: sunxiaoxiao@hdu.edu.cn. **侯文杰**，硕士研究生，主要研究领域为业务流程管理、流程监控预测等. **应钰柯**，硕士研究生，主要研究领域为业务流程管理、流程监控预测等. **俞东进**（**通信作者**），博士，教授，中国计算机学会（CCF）会员，主要研究领域为业务流程管理、大数据和软件工程等. E-mail：yudj@hdu.edu.cn.

prediction. To ensure that the inter-instance attributes defined in this paper have universality and can be applied in other BPs, the most common basic attributes are selected for the definition. Then, we thoroughly analyze and mine historical data of multiple process logs and select several key activity attributes that have significant impacts on execution time. The above two types of special attributes are combined with basic attributes that are directly obtained from event logs as the input of the prediction. Another research focus of this paper is to construct the prediction model to forecast the remaining execution time of an on-going instance, which is essentially a regression problem. To improve the ability and stability of the prediction model to be applied in more complex scenarios, the stacking technique is adopted here to fuse two machine learning models, i.e., LightGBM and XGBoost, into a two-layer hybrid model. Three experiments are performed in order to prove the performance of our method on four real-life datasets. The first experiment indicates that by adding the newly proposed two types of attributes, the Mean Average Error (MAE) of all datasets are decreased. And the two-layer hybrid models vastly outperform single-layer models on all four datasets, which prove the effectiveness of hybrid models. Therefore, in the second experiment, we further explore the performance of the two-layer hybrid model by conducting pair-wise fusion of LightGBM, XGBoost, Support Vactor Regression (SVR) and Bayesian Regression (BR). The results show that the combinations of LightGBM and XGBoost have the best prediction accuracy and when LightGBM is in the first layer and XGBoost is in the second layer (i.e., H-Model II), the performance is the best. In the third experiment, we perform comparisons of our H-Model II and the state-of-the-art methods, i.e., Long Short-Term Memory (LSTM) and single XGBoost. The results show that our hybrid model that considers attributes between instances and key activities outperforms LSTM and XGBoost by 11.6% and 15.8% on MAE, respectively.

Keywords Business Process Mining; Remaining Time Prediction; Machine Learning; Resource Competition; Fusion Model

1 引 言

如今，许多企业采用结构化形式（如日志）对其信息系统中的事件数据进行存储. 业务流程挖掘是一种从历史事件日志中提取有价值的知识的技术，利用这些知识可以帮助企业提高服务效率、服务速度和服务质量,从而增强其企业竞争力[1]. 近年来业务流程挖掘的重点不再局限于提供事件日志的脱机分析，而转向为业务流程优化提供在线支持. 其中预测性流程监控（PPM）[2]已成为业务流程挖掘的重要子领域之一，PPM 利用业务流程的历史执行日志生成预测模型进而预测正在执行的流程实例的一些可量化指标，例如剩余执行时间[3]、下一个要执行的活动[4]、流程执行结果[5]等. PPM 的目标在于提前预知未来可能出现的超时风险、流程偏离或执行失败,以便相关人员能够采取主动的纠正措施来降低流程风险并提高业务流程执行的质量和效率.

在 PPM 的任务中,业务流程剩余执行时间的预测对于正在进行的实例极其重要，它可以帮助相关人员调整实例的执行优先级，从而避免某些流程实例的执行超出时间期限[3]. 目前已开展的业务流程剩余执行时间预测的研究大多直接利用日志中记录的事件属性，未对其中的信息开展深入分析和挖掘[2]，多流程实例之间的竞争关系以及其他一些重要的信息数据往往被忽略[6].

为解决上述问题，本文对多个业务流程事件日志中的历史数据开展了深入的分析与挖掘，并提出对业务流程剩余时间预测具有重大影响的两类特征属性，即实例间属性和关键活动属性，分别表征多实例间的资源竞争关系和关键活动对剩余执行时间的影响. 同时，为提高预测模型在复杂场景下的应用性和稳定性，本文采用堆叠技术[7]将 LightGBM[8]和 XGBoost[9]两种学习模型结合为混合预测模型，并将本文提出的两类特征属性与直接从日志中提取的基本属性一起作为模型的输入进行剩余时间预测. 在四个真实数据集上开展的比对实验结果表明本文提出的考虑实例间属性和关键活动属性的混合预测模型在准确性上较其他模型有显著的提高.

本文的其余部分安排如下：第 2 节是与业务流程剩余执行时间预测相关工作的简要概述. 在第 3 节中介绍了研究相关概念的定义并详细讲解了本文提出的方法. 在第 4 节中，通过在四个真实数据集上进行的对比实验来分析本文提出方法的有效性. 最后，在第 5 节中对本文的研究工作进行了总结并阐述了未来工作展望.

2 相关工作

近几年,随着预测性流程监控技术的不断发展，涌现出大量有关业务流程剩余执行时间预测的研究工作。本文将这些方法按照是否进行流程感知划分为流程感知的预测方法与非流程感知的预测方法两类[2].

流程感知的预测方法首先从事件日志中抽象出业务流程结构，然后根据实例当前的输入信息预测其未来的全部执行行为和剩余执行时间. Aalst 等人[10]和 Polato 等人[11]选择带注释的变迁系统来表征业务流程结构. 其中 Aalst 等人[10]仅考虑事件日志中的控制流属性而未考虑其他属性如资源属性对未来转移行为的影响，为此 Polato 等人[11]综合考虑了更多事件日志中记录的特征属性，并使用贝叶斯分类器和支持向量机分别预测下一活动的状态和执行时间来实现对剩余时间的预测. Rogge-Solti 等人[12]将统计学的方法与变迁系统结合后对历史数据进行统计分析，模拟出变迁系统中每个状态的剩余时间分布，最后利用当前执行实例的状态来拟合分布进而预测剩余执行时间的分布情况，并采用抽样样本的平均时间作为实例的剩余时间预测值. Pandey 等人[13]将马尔科夫模型应用于业务流程的下一个执行活动的预测，并将该活动历史执行时间的均值作为其执行时间，从而实现对整个业务流程剩余时间的预测. Ceci 等人[14]使用传统的模式挖掘技术，首先从日志中找到其频繁序列模式构建局部模型，然后通过组合嵌套局部模型得到最终的预测模型.

非流程感知的预测方法主要利用机器学习技术（如决策树，循环神经网络等），这些方法依赖于隐式预测模型，并不需要关心流程实例未来的具体执行状况，因此其预测时间与流程感知预测的方法相比有较大的缩减. Tax 等人[4]将神经网络中的 LSTM 方法应用于预测性流程监控问题中，通过构建双层 LSTM 模型来分别预测执行的下一活动和时间属性，最终通过迭代实现对实例未来执行序列及剩余时间的预测. 该方法的主要问题在于当事件日志记录了大量循环结构时，预测未来执行轨迹会陷入事件循环，导致剩余时间预测值与真实执行时间相差较大. 不同于文献[4]需要洞察流程实例未来的每一步走向进而预测其剩余时间，Navarin 等人[15]使用 LSTM 模型直接对剩余时间进行预测，其预测耗费时间比文献[4]所需的时间更短，也不会陷入循环预测. 文献[4]和文献[15]均证明了 LSTM 在序列建模上具有较大优势，能捕捉长距离的依赖关系，获得精确的预测效果. 但 LSTM 也存在一些显著的缺点，如其对设备的计算能力要求较高、模型训练周期长、超参调试复杂. 此外，LSTM 属于黑盒模型，其可解释性较差. Márquez-Chamorro 等人[16]则提出了一种基于进化算法的学习模式，最终的预测模型由一系列决策规则组成，但其未考虑多实例之间的竞争关系.

综上，流程感知的预测方法由于需要依赖流程结构的显式表达，在非结构化业务流程的剩余时间预测上效果往往不佳，并有可能出现状态空间指数增长的状况；而非流程感知的预测方法无需依赖专家的专业知识并且能够使用机器学习等最新技术，在非结构化的业务流程预测上效果显著且效率更高，因此本文拟开展一种基于机器学习的非流程感知剩余时间预测研究. 此外，为解决目前大多数非流程感知方法存在因采用单一预测模型而在复杂应用场景下适应性较差的问题，本文使用堆叠技术将两个预测模型进行融合来构造双层混合预测模型，可有效提高模型的预测精度.

3 方法介绍

作为 BPM 在企业中的具体实践,业务流程驱动的信息系统（Process-Aware Information Systems，简称 PAIS）在企业中大量使用，以支持业务流程建模、设计、执行以及维护不同部门的员工灵活、高效地完成流程中的业务活动. PAIS 以日志的形式记录了业务流程执行时每个活动的相关信息（如活动的开始时间戳,活动名,执行资源等),如表 1 所示，这些信息可以很好地帮助相关人员了解业务流程的实际执行状态，同时也为本文的业务流程剩余时间预测研究提供了大量的真实数据.

3.1 问题定义

对于某业务流程来说，预测正在执行的流程实例的剩余时间，其本质是监督学习中的回归问题，即使用历史日志数据训练一个回归预测模型，然后将正在执行的流程实例的事件信息作为输入数据输入该模型，进而实现对该流程实例剩余执行时间的预测.

表 1 贷款金额受理流程事件日志片段

实例 ID	事件 ID	属性			
		活动名	开始时间戳	完成时间戳	执行资源
1	1	Registration	2014/04/02 16:00:48.000	2014/04/02 16:01:48.000	System
1	2	Acceptance of requests	2014/04/02 16:05:41.000	2014/04/02 16:18:23.000	group1
1	3	Collection of documents	2014/04/02 16:20:43.000	2014/04/02 17:47:58.000	group1
1	4	Completeness check	2014/04/02 17:55:18.000	2014/04/02 19:05:04.000	group2
2	5	Registration	2014/04/03 12:36:33.000	2014/04/02 12:37:48.000	System

回归问题的目标是给定已知的输入特征 x 和其对应的目标值 y，寻找一个输入特征 x 与其目标值 y 之间的映射函数 $y=g(x|\theta)$，其中 θ 是函数的参数，通常由训练学习得到. 当有新的数据 x' 出现时，可以通过该函数预测其相应的目标值 y'.

$$y'=g(x'|\theta) \tag{1}$$

因此业务流程剩余时间的预测问题即寻找一个最佳的映射函数（即回归预测模型）以实现当前已执行的事件信息与其剩余时间之间的映射，并使得损失函数的值最小.

3.2 基础概念

定义 1. 事件、轨迹. 事件日志中的每个事件 $e=(attr_1,attr_2,\ldots,attr_m)\in\mathcal{E}$ 都代表了流程中某一活动的一次执行，其中 $attr_i$ 表示事件拥有的属性（例如，每一个事件都有事件 ID 等属性），可以通过 $e.attr_i$ 获取事件相应属性的值；事件集 $\mathcal{E}$ 是日志中全部事件的集合. 流程轨迹 $\sigma=\langle e_1,\ldots,e_n\rangle$ 是由事件构成的有序序列，每条流程轨迹对应了业务流程的一次完整执行，即流程实例.

定义 2. 前缀轨迹. 前缀轨迹即流程轨迹 σ 的前 l 个事件构成的有序序列，其形式表达为 $PTr(\sigma,l)=\langle e_1,\ldots,e_l\rangle$，其中 l 表示截取事件在流程轨迹中的下标，也代表了前缀轨迹的长度.

定义 3. 活动集、资源集. 设 $\mathcal{A}=\{a_1,a_2,\ldots,a_n\}$ 为事件日志中全部活动名对应的单字符编码构成的集合，其中 a_i 是某一活动名对应的单字符编码，$\mathcal{R}=\{r_1,r_2,\ldots,r_m\}$ 为流程中所有活动的执行资源集合.

定义 4. 前缀活动序列. 前缀活动序列 $PAS(PTr(\sigma,l))$ 将前缀轨迹 $PTr(\sigma,l)$ 化为一个由活动名编码字符组成的字符串。例如，前缀轨迹 $PTr(\sigma_1,3)=\langle e_1,e_2,e_3\rangle$，且 e_1.activity=a_1，e_2.activity=a_2，e_3.activity=a_3，那么 $PAS(PTr(\sigma_1,3))=a_1a_2a_3$.

3.3 预测方法

业务流程事件日志中常见的属性有活动的执行资源、起始时间戳等，每个流程还拥有一些特有的属性，这些属性均可从事件日志中直接获取[2]. 通过对大量原始事件日志的数据分析，本文还挖掘出一些隐藏的特征属性，即实例间属性和关键活动属性.

3.3.1 实例间属性

在当前剩余时间预测相关的研究中，流程实例多被视为一个独立的个体，而忽略了与其一同执行的其他实例对该实例产生的影响. 如图 1 所示，实例 1 和实例 3 是具有多个相同活动的两个实例，但由于实例 3 开始执行的时间要早于实例 1，因此实例 3 能够更早地获取其所需的执行资源并完成其执行；而需要相同资源的实例 1 可能因为资源短缺而陷入停滞状态，进而增加其完成时间. 此外，在实际环境中，活动与资源之间的关系并非简单的一对一关系，而是复杂的多对多关系[17]，即使是不同类型的活动之间也存在着资源竞争. 因此将实例之间的关系考虑到实例剩余执行时间的预测中是非常有必要的.

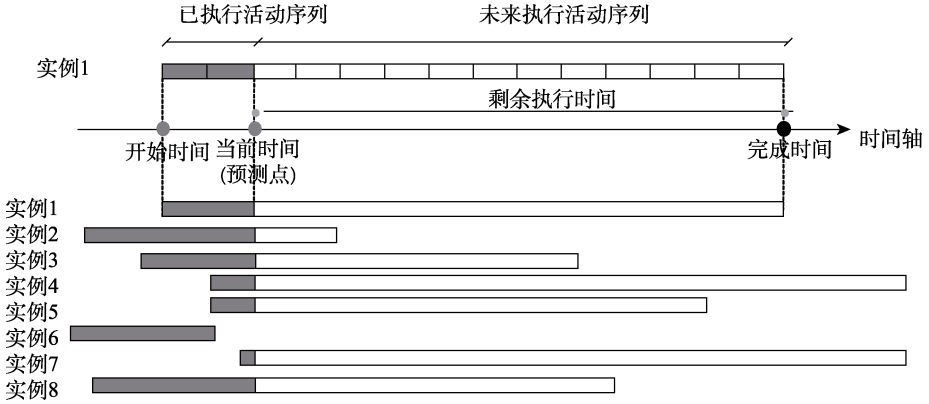

图 1 实例间竞争关系示例图（图中虚线表示时间线）

由于业务流程的执行具有一定的时间周期性，当前执行的活动在不久的将来极可能再次被执行，因此在预测点前特定时段内执行活动的数量分布可以一定程度上反映预测点后特定时段内执行活动的数量分布. 如果在某一时段内发生的实例数和事件数很多，则在这段时间内执行的实例之间的资源竞争也越激烈，对其剩余执行时间的影响也越大. 此外，为确保本文定义的实例间属性具有通用性，即可在其他业务流程中得到普遍应用，本文选取了事件日志中最常见的几个基本属性（即事件的时间戳，事件的活动名，事件的执行资源）来定义三类实例间属性，并一同将其作为预测输入属性.

定义 5. 区间实例数、区间事件数、区间事件加权数. 区间实例 $ExecCaseNum(t)$ 用来反映未来时间段 t 内可能执行的实例总数，其中 t 是采样的时间窗口大小，用于划定统计的时间范围，t_{point} 表示预测发生的时间点。区间事件数 $ExecEventNum(a_i,t)$ 用来反映未来时间段 t 内活动 a_i 对应事件的总数。区间事件加权数 $WeightEventNum(a_i,t)$ 综合考虑了与活动 a_i 对应事件具有竞争关系的所有事件，并通过统计历史日志中当前活动的不同执行资源的执行频率占比作为权重系数来计算当前活动的区间事件加权数，其公式定义如下：

$$ExecCaseNum(t)=\left|\left\{\sigma\left|(t_{point}-t)<\sigma.startTime<t_{point}\right.\right\}\right| \tag{2}$$

$$ExecEventNum(a_i,t)=\left|\left\{e\left|\left(\left(t_{point}-t\right)<e.startTime<t_{point}\right)\wedge\left(e.actvity=a_i\right)\right.\right\}\right| \tag{3}$$

$$WeightEventNum(a_i,t)=\sum_{\mathcal{A}}^{a_j}\frac{\left|prefR(a_i)\cap prefR(a_j)\right|}{\left|prefR(a_j)\right|}\times ExecEventNum(a_j,t) \tag{4}$$

其中 t 是采样的时间窗口大小，$prefR(a_i)$ 和 $prefR(a_j)$ 分别是可以执行活动 a_i 和活动 a_j 的所有资源的集合，$\left|prefR(a_i)\right|$ 和 $\left|prefR(a_i)\right|$ 分别表示可以执行活动 a_i 和活动 a_j 的资源的种类数。$prefR(a_i)\cap prefR(a_j)$ 是既能执行活动 a_i 又能执行活动 a_j 的资源的集合，$\left|prefR(a_i)\cap prefR(a_j)\right|$ 是既能执行活动 a_i 又能执行活动 a_j 的资源的种类数.

3.3.2　关键活动属性

本文通过对历史日志的深入挖掘分析发现，另一类对剩余执行时间影响较大的特征属性是关键活动属性，因为业务流程中不同活动的执行时间往往存在较大的差异。例如，活动 a_i 的平均执行时间为两天，而活动 a_j 的平均执行时间为两分钟，若 a_i 和 a_j 都是当前实例未来将要执行的活动，那么显然活动 a_i 对该实例剩余执行时间预测的影响要远大于活动 a_j.

为实现对关键活动的查找，本文实现了算法 1. 由于本文的研究目标是流程实例的剩余执行时间预测，因此在选取关键活动时，本文只对时间因素进行考虑，并采用活动执行时间的长短作为关键活动的判断标准. 算法 1 详细描述了关键活动的选取过程：首先，通过活动的开始时间戳属性和结束时间戳属性计算出每一个流程实例中每个活动的执行时间，并根据其执行时间对其进行降序排列. 然后根据这一排序结果为该流程实例中的每个活动分配一个 rank 值，其中执行时间最久的活动将分配到 1，其余活动根据其排序结果 rank 值逐次增加 1. 最后统计整个历史日志中不同活动的平均 rank 值，并选取平均 rank 值较小的 K 个活动作为该业务流程的关键活动. 根据二八定律[18]本文选取 k%为 20%.

算法 1.　查找关键活动.

输入：historical event log $\mathcal{L}$　//历史日志

输出：*KeyActivitiesSet*　//关键活动集合

FOREACH Trace σ in $\mathcal{L}$ **DO**
　$headnode=\sigma.getFirstEvent()$;
　$tailnode=\sigma.getLastEvent()$;
　$\sigma.exectime=tailnode.endTime-headnode.startTime$;
　FOREACH event e in σ **DO**
　　$e.exectime=e.endTime-e.startTime$;
　ENDFOREACH
　$SortEventByExecTime(\sigma)$;
　$currentEventRank=1$;
　FOREACH event e in σ **DO**
　　$e.rank=currentEventRank$;
　　$currentEventRank=currentEventRank+1$;
　ENDFOREACH
ENDFOREACH
FOREACH *activity* a in $\mathcal{A}$ **DO**
　$a.avgRank=CaluAvgRank(a)$;
ENDFOREACH
$K=round(k\%\times|\mathcal{A}|)$
KeyActivitiesSet = selectTopKavgRank($\mathcal{A},K$**)** //选择 K 个平均 rank 值最小的活动作为关键活动

在历史流程中找到关键活动后，本文设计了算

法 2，用于计算剩余关键活动近似数（*ANRKA*）属性，以作为整个剩余时间预测的输入之一. 算法大致分为三步：

步骤 1：根据前文中前缀轨迹和前缀活动序列的定义，将历史日志中完整且已经执行结束的轨迹衍生出不同执行长度的前缀轨迹，例如长度为 3 的轨迹 $\sigma=\langle e_1,e_2,e_3\rangle$，其可以衍生出长度为 1 和 2 的前缀轨迹 $PTr(\sigma,1)=\langle e_1\rangle$ 和 $PTr(\sigma,2)=\langle e_1,e_2\rangle$ 并且其对应的剩余执行时间和剩余关键活动数均已知. 然后，根据衍生的前缀轨迹获取全部前缀活动序列并组成前缀活动序列集 *historySeqSet*. 随后获取当前要预测的前缀轨迹 $PTr(\sigma,l)$ 所对应的活动序列 *currentSeq*，其长度为 l. 从 *historySeqSet* 中获取与其相同长度的前缀活动序列构成子集 *subSeqSet*. 对于 *subSeqSet* 中的每一个序列，依次计算其与 *currentSeq* 的相似度. 本文选用了莱温斯坦距离（Levenshtein Distance）[19] 来计算两个序列之间的相似度. 最终，从长度为 l 的子集中找到了与 *currentSeq* 最相似的前缀活动序列 *currentSeq*，他们之间的相似度为 *bestSimilarity*.

步骤 2：在步骤 1 中已找到与 *currentSeq* 最相似且长度同为 l 的相似序列 *similarSeq*. 接下来，算法将要扩展搜索范围. 由于 *currentSeq* 和 *similarSeq* 之间的相似度为 *bestSimilarity*，这意味着将 *similarSeq* 转变为 *currentSeq* 所需要的最小单字符编辑操作数（即插入、删除或替换）是 *bestSimilarity*. 因此，在长度范围为[*l*- *bestSimilarity*，*l*+ *bestSimilarity*]的前缀活动序列集中，有可能会找到一个与 *currentSeq* 更相似的前缀活动序列. 因此，重新开始步骤 1 的搜索过程，并将前缀活动序列的子集所包含的前缀活动序列的长度范围扩大到[*l*- *bestSimilarity*，*l*+ *bestSimilarity*]，最终找到与 *currentSeq* 最相似的前缀活动序列.

步骤 3：在得到最相似的前缀活动序列之后，统计该前缀活动序列的剩余关键活动数. 值得注意的是一条前缀活动序列可能对应有多条不同的前缀轨迹，此时计算它的平均剩余关键活动数作为当前要预测轨迹的 *ANRKA* 值.

算法 2. 计算剩余关键活动近似数.

输入：historical event log $\mathcal{L}$, current prediction trace $PTr(\sigma,l)$

输出：$ANRKA(PTr(\sigma,l))$

$historySeqSet=getPAS(\mathcal{L})$;
$currentSeq=PAS(PTr(\sigma,l))$;
$subSeqSet=getSubSet(historySeqSet,l,l)$;
$bestSimilarity=infinity$;
FOREACH sequence *seq* in *subSeqSet* **DO**
  $tmpSimilarity=calcSimilar(currentSeq,seq)$;
  IF $tmpSimilarity<bestSimilarity$ **DO**
    $bestSimilarity=tmpSimilarity$;
    $similarSeq=seq$;
  ENDIF
ENDFOREACH
$range=bestSimilarity$;
$subSeqSet1=getSubSet(historySeqSet,l-range,l-l)$;
$subSeqSet2=getSubSet(historySeqSet,l+1,l+range)$;
$subSeqSet3=subSeqSet1\cup subSeqSet2$;
FOREACH sequence *seq* in *subSeqSet*3 **DO**
  $tmpSimilarity=calcSimilar(currentSeq,seq)$;
  IF $tmpSimilarity<bestSimilarity$ **DO**
    $bestSimilarity=tmpSimilarity$;
    $similarSeq=seq$;
  ENDIF
ENDFOREACH
$ANRKA(PTr(\sigma,l))=getAvgRKA(\mathcal{L},similarSeq)$

3.3.3 双层混合模型

由于业务流程的应用场景复杂多样，在一些应用场景下，单一回归模型的预测结果往往不太理想，为此本文使用堆叠技术对 XGBoost 和 LightGBM 两个机器学习模型进行融合，得到一个效果较优的混合预测模型。其中两个单一预测模型 XGBoost、LightGBM 在其他领域已被证实能取得较好的预测效果[20,21]. 双层混合模型的构建如图 2 所示，其训练步骤如下：

（1）训练集分割：将原始训练集等分为两个子集：训练集 1、训练集 2.

（2）训练第一层模型：将第一步得到的两个训练集分别输入到 XGBoost 和 LightGBM 的训练模块中进行模型的学习训练. 使用交叉验证和网格搜索的方法得到最佳的预测模型并保存到预测模块中.

（3）合成新训练集：为提高预测效果，使用第一层模型中的预测模块对原始训练数据进行预测，并将预测结果与原始训练数据整合形成新的训练数据. 具体地，将训练集 2 输入到 XGBoost 的预测模块中得到的结果集与原始的训练集 2 整合得到训练集 3，将训练集 1 输入到 LightGBM 的预测模块中得到的结果集与原始的训练集 1 整合得到训练集 4.

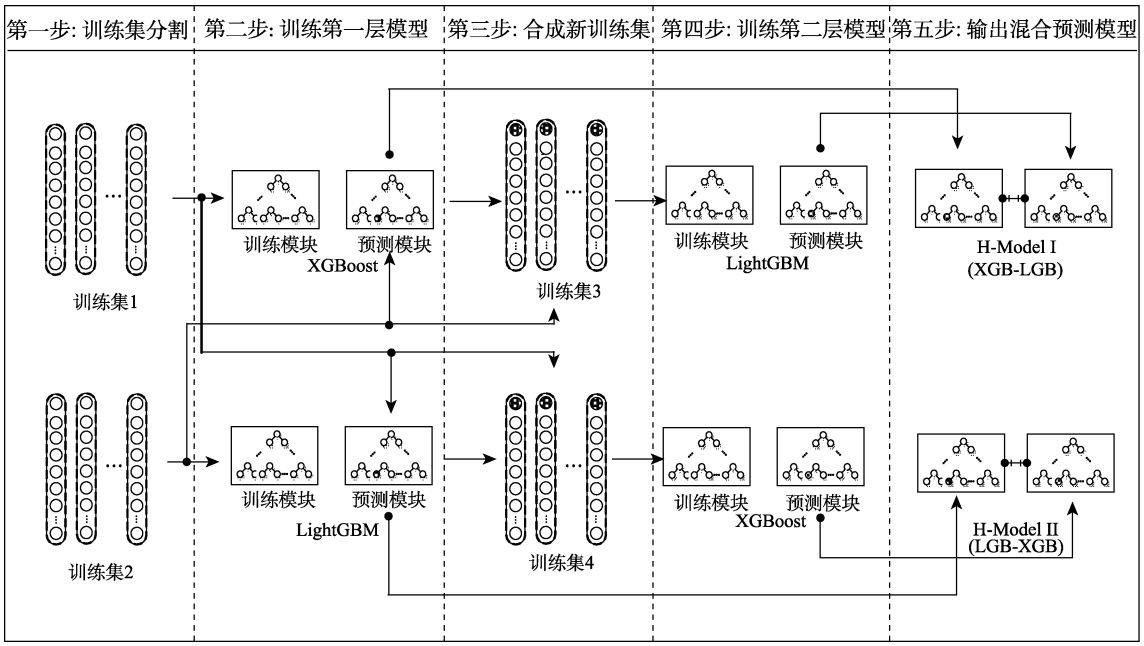

图 2　双层混合模型训练图

（4）训练第二层模型：将上一步得到的合成训练集 3、4 分别输入到 LightGBM 和 XGBoost 的训练模块中进行二次学习训练。使用交叉验证和网格搜索的方法得到最佳的预测模型并保存到预测模块中.

（5）输出混合预测模型：将第 2 步和第 4 步中得到的预测模块进行整合形成最终的混合预测模型. 其中，混合预测模型 H-Model I 的第一层模型是 XGBoost，第二层模型是 LightGBM，混合预测模型 H-Model II 的第一层模型是 LightGBM，第二层模型是 XGBoost.

4　相关实验及分析

4.1　实验环境

本文实验配置如下：操作系统：Windows10 专业版 64 位；处理器：英特尔酷睿 i5-6500(3.30GHz，四核四线程)；XGBoost 版本：0.90；LightGBM 版本：2.2.3.

4.2　数据处理

4.2.1　数据集

本文选取了四个真实的业务流程的历史日志作为验证实验的数据集，其来源于 4TU Centre for Research Data(https://data.4tu.nl/repository). 四个数据集的相关信息如下所示：

Production：该数据集包含了某制造公司 2012 年 1 月至 3 月部分产品的制造流程信息. 每条流程实例详细生产相关的活动、工人、机器等信息 (https://data.4tu.nl/repository/uuid:68726926-5ac5-4fab-b873-ee76ea412399).

Helpdesk：该数据集包含了意大利某软件公司服务台 2010 年 1 月至 2014 年 1 月的票务管理流程执行信息 (https://data.4tu.nl/repository/uuid:0c60edf1-6f83-4e75-9367-4c63b3e9d5bb).

BPIC2012：该数据集来源于荷兰某财政机构，记录了 2011 年 10 月至 2012 年 3 月内该机构的贷款申请流程执行信息 (https://data.4tu.nl/repository/uuid:3926db30-f712-4394-aebc-75976070e91f).

BPIC2017：该数据集与 BPIC2012 来源于同一财政机构的申请贷款流程，记录了该机构 2016 年 1 月至 2017 年 2 月的贷款申请流程执行信息. 不同于 BPIC2012，BPIC2017 中相关活动的信息更加详细丰富(https://data.4tu.nl/repository/uuid:3926db30-f712-4394-aebc-75976070e91f).

四个数据集的相关统计信息如表 2 所示.

表 2　事件日志统计分析数据表

日志名称	实例总数	事件总数	活动类别数	资源类别数	轨迹长度范围
Production	225	4544	55	31	1-175
Helpdesk	4580	21348	14	22	1-15
BPIC2012	13087	262200	36	63	1-175
BPIC2017	31509	1202267	26	147	1-180

4.2.2　数据预处理

在开始正式的实验之前，需要对原始的数据进

行几步预处理操作.

（1）数据集分割. 为还原真实应用场景，所有案例根据其发生时间进行排序，并选取前面发生的80%案例数据作为训练集，剩余的20%作为测试集.

（2）特征提取及编码. 由于不同的流程实例的活动执行时间往往存在较大差异，为此本文选定了多个时间窗口 t，分别为一小时、六小时、一天、一周、一个月以及距当前实例开始的时间间隔，来生成每个事件的多个实例间属性并将其一同作为预测输入特征之一. 训练模型会自动学习其中对剩余执行时间预测影响较大的特征并调整权重系数. 然后根据算法 1 和算法 2，为每个样本找到代表其关键活动属性的 *ANRKA* 值. 同时将原始日志中直接记载的基本属性根据其特征值是否为数值类型划分为数值型特征和类别型特征，其中类别型特征需要对其进行 one-hot 编码.

（3）数据清洗. 对于训练集中的缺失值，使用该项属性值的中位数进行填充. 对于训练集中的异常数据，则选用传统的箱线图[22]进行异常值的划定和平滑：首先找到每项特征属性值的第一四分位数 Q_1 和第三四分位数 Q_3，计算出该特征属性值的四分位数全距 IQR，以及上限值 $\text{Limit}_{\text{upper}}$ 和下限值 $\text{Limit}_{\text{lower}}$；然后依次遍历该项属性的所有特征值，将大于上限值的特征值使用上限值替换，小于下限值的使用下限值替换. 其计算公式如下：

$$IQR = Q_3 - Q_1 \tag{5}$$

$$Limit_{upper} = Q_3 + 1.5 \times IQR \tag{6}$$

$$Limit_{lower} = Q_1 - 1.5 \times IQR \tag{7}$$

4.3 实验结果分析

4.3.1 衡量指标

由于同一流程中不同实例之间的剩余时间差别显著，并且在流程逼近结束时，其剩余执行时间的值可能很小（接近零），因此本文选择了平均绝对误差（MAE）和均方根误差（RMSE）作为本次实验预测效果的衡量指标. 其计算公式如下:

$$R_{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|y(i)_{predictTime} - y(i)_{realTime}\right| \tag{8}$$

$$R_{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y(i)_{predictTime} - y(i)_{realTime})^2} \tag{9}$$

其中 $y(i)_{realTime}$ 代表真实的剩余执行时间，$y(i)_{predictTime}$ 是预测模型得到的预测时间. R_{MAE} 和 R_{RMSE} 的值越小，说明预测得越精准，预测模型的效果越好.

4.3.2 结果分析

为验证本文提出的两类特征属性以及混合预测模型的可行性，本文在四个真实数据集进行了多次实验. 对于每个数据集，本文为模型训练生成了四种不同的数据样本. 其中第一种数据样本（Basic）只包含事件日志中的基本属性，这些属性在日志中有直接记载；第二种数据样本（Int-Ins）在第一种数据样本的基础上结合了本文提出的实例间属性；第三种数据样本（K-Act）则是第一种数据样本与本文提出的关键活动属性相结合的结果；第四种数据样本（All）在第一种数据样本的基础上同时集成了实例间属性和关键活动属性. 此外，为证明本文提出的双层混合模型对预测结果的提升效果，实验选择两个单一模型作为对比实验，即 S-Model I 和 S-Model II 分别选用了 LightGBM 和 XGBoost 作为训练模型，实验结果如图 3 所示. 由图中可知，在四个数据集上双层混合预测模型的预测精度均优于单一模型，这证实了本文提出的混合预测模型对预测精度的有效提升. 值得注意的是，H-Model II 的预测效果比 H-Model I 的效果略好，这是由于 LightGBM 采用了直方图策略对连续特征值进行了离散化，进而对预测精度产生了一定影响，因而将其作为第二层的预测输出模型时效果略差. 但这一策略的优势在于其能大大节省读入数据所占用的内存，并在一定程度上起到防止过拟合的效果，因而在处理海量数据时，其训练周期更短，且能保持预测精度[8,23]. 同时图中还可以观察到，在基础属性中加入任意一种本文提出的特征属性之后，预测的效果均能得到一定的提升. 尤其是当本文提出的两类特征属性（实例间属性和关键活动属性）一同被加入输入样本时，提升效果更加显著.

图 4 显示了每个数据集中不同前缀长度轨迹的预测结果，其中输入的数据样本选用了第四类数据样本，因为图 3 的实验已经证实该类数据样本的预测结果最佳. 由图 4 可以发现，当轨迹前缀长度偏小时，预测的精度会随着前缀长度的增加而增大，这是因为随着前缀长度的增加，获取的已执行信息越多并且剩余执行时间也越短，因此更加容易预测. 但当前缀长度过长时，由于日志中拥有相同长度的数据样本量骤减，因此模型训练阶段学习样本过少，从而导致预测模型的预测精度急剧下降. 同时由于实例的执行接近结束，其剩余时间也近似于 0，因此容易出现偏差过大的现象.

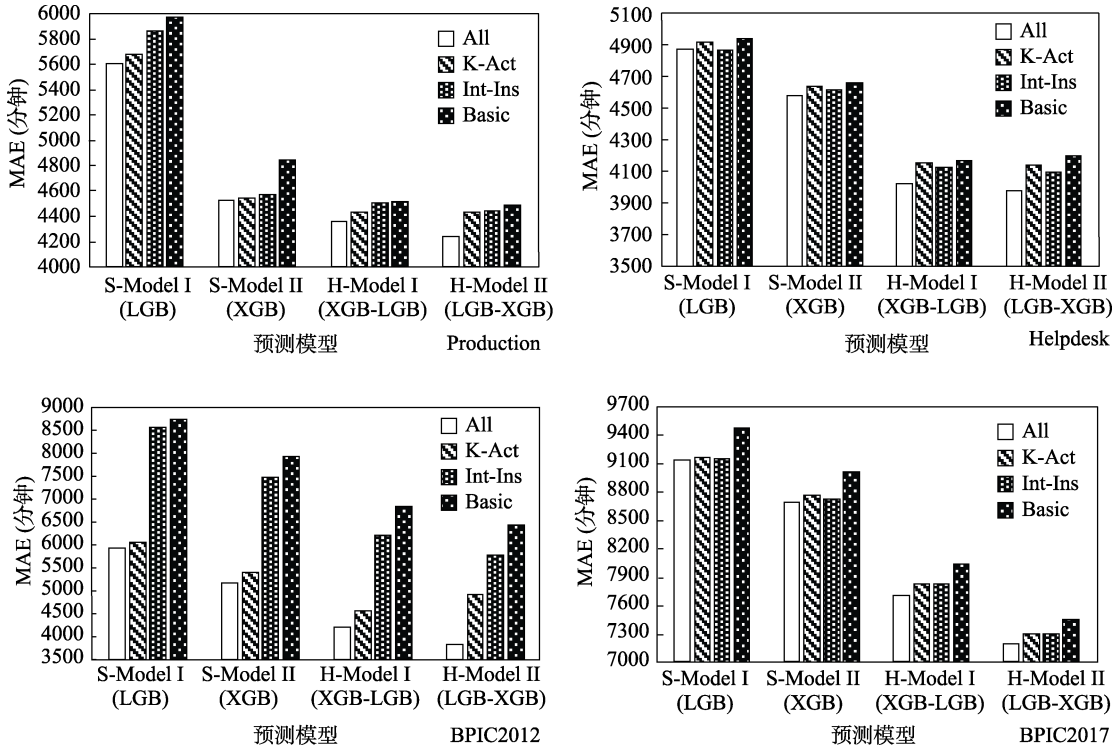

图 3　流程剩余时间预测精度结果展示图

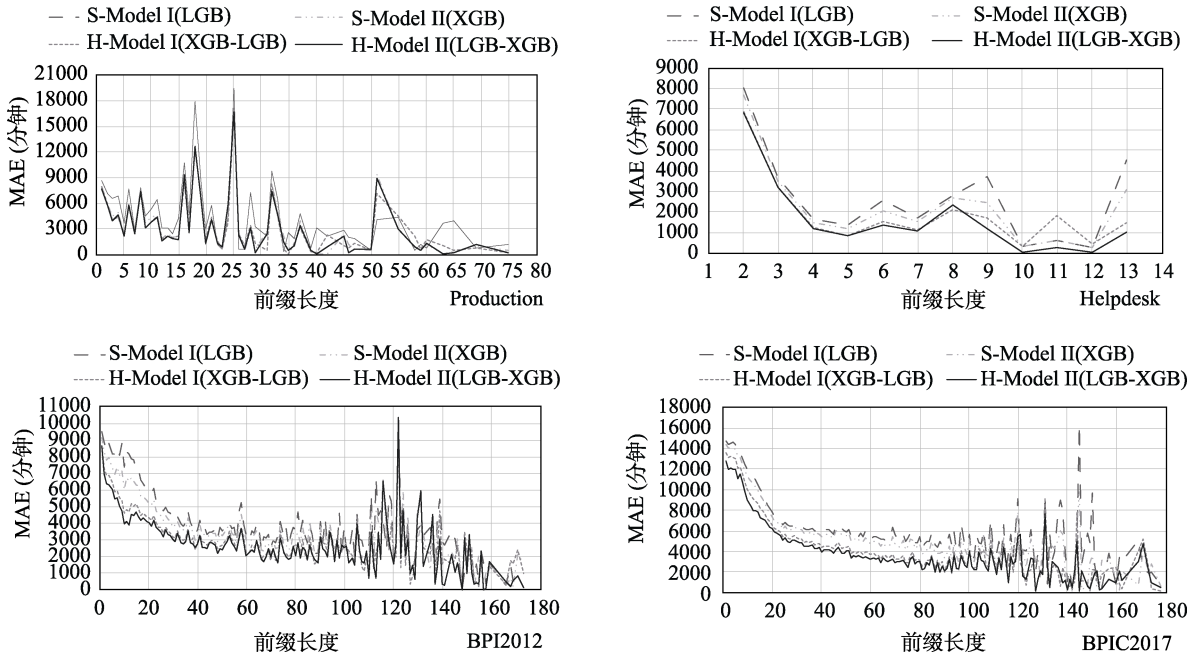

图 4　不同前缀长度的轨迹预测效果展示图

为了进一步证明本文提出的双层混合模型的有效性，本文分别对 XGBoost 模型（XGB）、LightGBM 模型（LGB）、支持向量回归模型（SVR）和贝叶斯岭回归模型（BR）这四种回归模型进行两两组合来训练混合模型，实验结果如表 3 所示，其中最优结果使用粗体标记. 由表可见，由 XGBoost 模型和 LightGBM 模型构建的混合模型 XGB-LGB（H-Model I）和 LGB-XGB（H-Model II）效果最佳. 究其原因，首先混合模型的预测效果与构建它的两个单层模型紧密相关，而 XGBoost 和 LightGBM 均为提升树改进算法，具有泛化能力强、支持并行化学习、防止过拟合等优点. 它们对目标函数使用泰勒公式进行二阶展开使得模型可以学习得更加充分，能够有效提高预测精度. 此外，由于双层模型的第一层模型

会对剩余时间进行初步的预测，第二层模型则会结合第一层模型的预测结果和已有的数据特征属性进行再次学习，相当于对第一次预测结果的一次修正，因此其预测效果较普通模型更佳.

最后，为进一步证明本文提出方法的有效性，我们选取了文献[3]中几类剩余时间预测方法中效果最优的 LSTM 算法[15]和效果次优的 XGBoost 算法[6]作为本文的对比算法进行对比实验. 在本次实验中，仍选用效果最佳的第四种数据样本作为输入，并选择了混合预测模型中综合表现最佳的 H-Model II 模型与其他算法进行对比，实验结果如表 4 所示，H-Model II 在所有数据集中的表现均优于 LSTM 和 XGBoost，其主要原因在于混合预测模型的第二层模型会对第一层的学习结果进行二次学习，能够在一定程度上弥补第一层模型的部分缺陷，进而提高了预测结果.

总体而言，图 3 和图 4 说明了本文提出的两种隐藏特征属性对剩余时间预测的重要性，并且证明本文提出的双层混合预测模型可以有效地提高预测结果的准确性. 表 3 和表 4 的结果则表明在四个真实数据集中，本文方法相较其他混合模型和已有的其他算法均具有更好的预测性能.

表 3 混合模型的预测结果对比表

衡量指标	混合模型	数据集			
		Production	Helpdesk	BPIC2012	BPIC2017
MAE	XGB-SVR	6220.49	6388.87	6603.16	9234.74
	LGB-SVR	6077.82	6663.06	6583.80	9912.79
	XGB-BR	6223.71	6184.20	6736.54	9401.94
	LGB-BR	6086.77	6104.62	6631.65	9991.82
	SVR-LGB	6733.99	5760.91	5906.91	10269.13
	SVR-XGB	6741.72	5700.97	5646.24	10230.49
	BR-LGB	6656.68	5695.55	5940.81	10324.55
	BR-XGB	6666.17	5764.17	5701.20	10238.24
	XGB-LGB (H-Model I)	4363.49	4022.91	4213.17	7701.94
	LGB-XGB (H-Model II)	**4244.75**	**3977.83**	**3850.65**	**7189.11**
RMSE	XGB-SVR	11213.26	9153.88	12160.64	14175.70
	LGB-SVR	11074.12	10305.07	12719.65	16003.66
	XGB-BR	11220.53	9053.54	12409.08	14267.76
	LGB-BR	11086.91	10248.10	12799.44	16032.09
	SVR-LGB	12368.38	8894.02	8371.59	14056.35
	SVR-XGB	12241.95	8902.73	7975.61	13983.78
	BR-LGB	11693.20	8813.88	8371.96	13916.29
	BR-XGB	12351.96	8912.74	8016.43	13991.69
	XGB-LGB (H-Model I)	**9925.69**	**7382.71**	7857.17	12152.82
	LGB-XGB (H-Model II)	10109.03	7501.15	**7558.74**	**11473.72**

表 4 三种不同剩余时间预测算法的对比结果表

衡量指标	预测方法	数据集			
		Production	Helpdesk	BPIC2012	BPIC2017
MAE	LSTM	4557.47	4582.80	5195.90	8689.28
	XGBoost	4920.90	4745.76	5337.26	9140.35
	LGB-XGB (H-Model II)	**4363.49**	**4022.91**	**4213.17**	**7701.94**
RMSE	LSTM	14806.23	8601.52	8134.03	16030.47
	XGBoost	10239.21	7733.40	8810.10	13184.92
	LGB-XGB (H-Model II)	**10109.03**	**7501.15**	**7558.74**	**11473.72**

5 结　语

为解决业务流程的剩余时间预测问题，本文挖掘了两类对剩余时间影响较大的隐藏特征属性，即实例间属性和关键活动属性，其分别考虑了实例之间的资源竞争以及不同活动对流程执行时间的影响. 本文将上述两类特征属性与日志中记载的基本属性相结合构成了预测的输入. 此外，本文提出了一种使用堆叠技术融合 XGBoost 和 LightGBM 模型构建的双层混合预测模型，可有效提高模型预测的精度和在复杂应用场景中的适应性. 通过在四个真实的数据集上的对比实验可知，本文提出的考虑了实例间属性以及关键活动属性的混合预测模型比以往的预测方法具有更好的效果.

此外，本文提出的双层混合预测模型基于机器学习技术，不需要依赖于业务流程模型，因此能很好地适用于非结构化的业务流程，如医院诊断流程、银行审批流程等执行走向依赖于人为判断的流程. 此外，一般混合模型的不足之处在于其训练周期多长于单层模型，而本文提出的混合预测模型由 XGBoost 模型和 LightGBM 模型两个模型构成，它们均支持并行运算，因此能有效减少模型的整体训练周期. 然而，与目前已有的其他剩余时间预测方法类似，该模型暂时无法较好地应用于对剩余时间预测精度要求极高的业务流程.

在未来的工作中，我们计划将对业务流程中影响人力资源执行效率的相关因素展开进一步的考虑（如员工的工作负载、员工的技能熟练度、员工的搭配等），这些因素也会对流程的剩余执行时间产生较大影响. 对于混合预测模型，我们还计划继续尝试将其他机器学习方法一同集成到其中，并寻找最优的组合模型.

参　考　文　献

[1] Aalst W M P. Process Mining: Data Science in Action. New York, USA: Springer Publishing Company, 2016

[2] Marquez-Chamorro A E, Resinas M, Ruiz-Cortes A. Predictive monitoring of business processes: a survey. IEEE Transactions on Services Computing, 2018, 11(06): 962-977

[3] Verenich I, Dumas M, Rosa M L, et al. Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. ACM Transactions on Intelligent Systems and Technology, 2019, 10(4): 1-34

[4] Tax N, Verenich I, La Rosa M, et al. Predictive business process monitoring with LSTM neural networks//Proceedings of the International Conference on Advanced Information Systems Engineering. Essen, Germany, 2017: 477-492

[5] Teinemaa I, Dumas M, Rosa M L, et al. Outcome-oriented predictive process monitoring: review and benchmark. ACM Transactions on Knowledge Discovery from Data, 2019, 13(2): 1884-2021

[6] Senderovich A, Di Francescomarino C, Ghidini C, et al. Intra and inter-case features in predictive process monitoring: A tale of two dimensions//Proceedings of the International Conference on Business Process Management. Barcelona, Spain, 2017: 306-323

[7] Wolpert D H. Stacked generalization. Neural networks, 1992, 5(2): 241-259

[8] Ke G, Meng Q, Finley T, et al. Lightgbm: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems. 2017: 3146-3154

[9] Chen T, Guestrin C. Xgboost: a scalable tree boosting system//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, USA, 2016: 785-794

[10] Van Der Aalst W M P, Schonenberg M H, Song M. Time prediction based on process mining. Information Systems, 2011, 36(2): 450-475

[11] Polato M, Sperduti A, Burattin A, et al. Time and activity sequence prediction of business process instances. Computing, 2018, 100(9): 1005-1031

[12] Rogge-solti A, Weske M. Prediction of remaining service execution time using stochastic petri nets with arbitrary firing delays//Proceedings of the International Conference on Service-Oriented Computing. Koloa, USA, 2013: 389-403

[13] Pandey S, Nepal S, Chen S. A test-bed for the evaluation of business process prediction techniques//Proceedings of the 7th International Conference on Collaborative Computing: Networking, Applications and Worksharing. Orlando, USA, 2011: 382-391

[14] Ceci M, Lanotte P F, Fumarola F, et al. Completion time and next activity prediction of processes using sequential pattern mining// Proceedings of the International Conference on Discovery Science. Bled, Slovenia, 2014: 49-61

[15] Navarin N, Vincenzi B, Polato M, et al. LSTM networks for data-aware remaining time prediction of business process instances//Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence. Honolulu, USA, 2017: 1-7

[16] Marquez-Chamorro A E, Resinas M, Ruiz-Cortes A, et al. Run-time prediction of business process indicators using evolutionary decision rules. Expert Systems with Applications, 2017, 87: 1-14

[17] Xie Y, Chien C F, Tang R Z. A dynamic task assignment approach based on individual worklists for minimizing the cycle time of business processes. Computers & Industrial Engineering, 2016, 99: 401-414

[18] Chen Y S, Chong P P, Tong M Y. Mathematical and computer modelling of the Pareto principle. Mathematical and Computer Modelling, 1994, 19(9): 61-80

[19] Levenshtein V I. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady. 1966, 10(8): 707-710

[20] Babajide Mustapha I, Saeed F. Bioactive molecule prediction using extreme gradient boosting. Molecules, 2016, 21(8): 983

[21] You Z H, Zhan Z H, Li L P, et al. Accurate prediction of ncrna-protein interactions from the integration of sequence and evolutionary information. Frontiers in Genetics, 2018, 9: 458

[22] Sim C H, Gan F F, Chang T C. Outlier labeling with boxplot procedures. Journal of the American Statistical Association, 2005, 100(470): 642-652

[23] Fu F, Jiang J, Shao Y, et al. An experimental evaluation of large scale gbdt systems. Proceedings of the VLDB Endowment, 2019, 12(11): 1357-1370

SUN Xiao-Xiao, Ph. D., lecturer. Her current research interests include business process management, spatio-temporal data mining, etc.

HOU Wen-Jie, M. S. candidate. His current research interests include business process management, process monitoring, etc.

YING Yu-Ke, M. S. candidate. Her current research interests include business process management, process monitoring, etc.

YU Dong-Jin, Ph. D., professor. His current research interests include business process management, big data, software engineering, etc.

Background

Process mining is a technique that dedicates to extracting process-related knowledge from redundant event logs and providing profound insight for stakeholders. One essential application of process mining is predictive process monitoring, which is able to provide timely predictive information such as the remaining execution time, the next activity to be executed or the final execution outcome according to executed information of the current instance and historical event logs. Among tasks of predictive process monitoring, remaining time prediction of processes is the most important one for the on-going instance as it helps the relevant personnel to adjust the priority of activities, thus avoiding the execution of some process instances exceeding their deadlines. A variety of remaining time prediction methods have been put forward to assist stakeholders' decision-making during the past decade. Especially in recent years, machine learning techniques have been applied to predict the remaining time of business processes, and they are proven to outperform existing methods. However, most of the current researches on remaining time prediction consider feature attributes that are directly extracted from logs as the input of the prediction without analyzing or mining the logs in depth. Moreover, the competition between multiple process instances executed together are usually ignored.

To address the above problems, we analyze and mine the historical data of multiple process logs, and then characterize several features attributes to represent resource competition and prioritize several key activities that strongly impact remaining execution time. The two types of feature attributes are combined with normal feature attributes derived from logs as the input of the prediction. In addition, we transform the remaining time prediction problem into a supervised learning problem in this paper, and the stacking technique is adopted to fuse two machine learning models (e.g., XGBoost and LightGBM) into a hybrid model to improve the ability and stability of the prediction model in more complex scenarios.

This work is supported by National Natural Science Foundation, China (No.61472112), Zhejiang Provincial Natural Science Foundation of China (LQ20F020017), and the Zhejiang Provincial Key Science and Technology Foundation, China (No.2020C01165、No.2017C01010).