

# 一种预测流程剩余时间的可解释特征分层方法\*

郭娜<sup>1</sup>, 刘聪<sup>2,3</sup>, 李彩虹<sup>2</sup>, 陆婷<sup>1</sup>, 闻立杰<sup>4</sup>, 曾庆田<sup>3</sup>



<sup>1</sup>(山东理工大学 电气与电子工程学院, 山东 淄博 255000)

<sup>2</sup>(山东理工大学 计算机科学与技术学院, 山东 淄博 255000)

<sup>3</sup>(山东科技大学 计算机科学与工程学院, 山东 青岛 266590)

<sup>4</sup>(清华大学 软件学院, 北京 100084)

通信作者: 刘聪, E-mail: [liucongchina@163.com](mailto:liucongchina@163.com)

**摘要:** 流程剩余时间预测对于业务异常的预防和干预有着重要的价值和意义. 现有的剩余时间预测方法通过深度学习技术达到了更高的准确率, 然而大多数深度模型结构复杂难以解释预测结果, 即不可解释问题. 此外, 剩余时间预测除了活动这一关键属性还会根据领域知识选择若干其他属性作为预测模型的输入特征, 缺少通用的特征选择方法, 对于预测的准确率和模型的可解释性存在一定的影响. 针对上述问题, 提出基于可解释特征分层模型 (explainable feature-based hierarchical model, EFH model) 的流程剩余时间预测框架. 具体而言, 首先提出特征自选择策略, 通过基于优先级的后向特征删除和基于特征重要性值的前向特征选择, 得到对预测任务具有积极影响的属性作为模型输入. 然后提出可解释特征分层模型架构, 通过逐层加入不同特征得到每层的预测结果, 解释特征值与预测结果的内在联系. 采用 LightGBM (light gradient boosting machine) 和 LSTM (long short-term memory) 算法实例化所提方法, 框架是通用的, 不限于选用算法. 最后在 8 个真实事件日志上与最新方法进行比较. 实验结果表明所提方法能够选取有效特征, 提高预测的准确率, 并解释预测结果.

**关键词:** 流程挖掘; 剩余时间预测; 特征选择; 可解释; 分层模型

**中图法分类号:** TP311

中文引用格式: 郭娜, 刘聪, 李彩虹, 陆婷, 闻立杰, 曾庆田. 一种预测流程剩余时间的可解释特征分层方法. 软件学报, 2024, 35(3): 1341–1356. <http://www.jos.org.cn/1000-9825/6824.htm>

英文引用格式: Guo N, Liu C, Li CH, Lu T, Wen LJ, Zeng QT. Explainable Feature-based Hierarchical Approach to Predict Remaining Process Time. Ruan Jian Xue Bao/Journal of Software, 2024, 35(3): 1341–1356 (in Chinese). <http://www.jos.org.cn/1000-9825/6824.htm>

## Explainable Feature-based Hierarchical Approach to Predict Remaining Process Time

GUO Na<sup>1</sup>, LIU Cong<sup>2,3</sup>, LI Cai-Hong<sup>2</sup>, LU Ting<sup>1</sup>, WEN Li-Jie<sup>4</sup>, ZENG Qing-Tian<sup>3</sup>

<sup>1</sup>(School of Electrical and Electronic Engineering, Shandong University of Technology, Zibo 255000, China)

<sup>2</sup>(School of Computer Science and Technology, Shandong University of Technology, Zibo 255000, China)

<sup>3</sup>(College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China)

<sup>4</sup>(School of Software, Tsinghua University, Beijing 100084, China)

**Abstract:** Remaining process time prediction is important for preventing and intervening in abnormal business operations. For predicting the remaining time, existing approaches have achieved high accuracy through deep learning techniques. However, most of these techniques involve complex model structures, and the prediction results are difficult to be explained, namely, unexplainable issues. In addition, the

\* 基金项目: 国家自然科学基金 (61902222); 山东省泰山学者工程专项基金 (ts20190936, tsqn201909109); 山东省自然科学基金优秀青年基金 (ZR2021YQ45); 山东省高等学校青创科技计划创新团队项目 (2021KJ031); 山东科技大学领军人才与优秀科研团队计划 (2015TDJH102)

收稿时间: 2022-07-18; 修改时间: 2022-09-12; 采用时间: 2022-10-05; jos 在线出版时间: 2023-05-10

CNKI 网络首发时间: 2023-05-11

prediction of the remaining time usually uses the key attribute, namely activity, or selects several other attributes as the input features of the predicted model according to the domain knowledge. However, a general feature selection method is missing, which may affect both prediction accuracy and model explainability. To tackle these two challenges, this study introduces a remaining process time prediction framework based on an explainable feature-based hierarchical (EFH) model. Specifically, a feature self-selection strategy is first proposed, and the attributes that have a positive impact on the prediction task are obtained as the input features of the model through the backward feature deletion based on priority and the forward feature selection based on feature importance. Then an EFH model is proposed. The prediction results of each layer are obtained by adding different features layer by layer, so as to explain the relationship between input features and prediction results. The study also uses the light gradient boosting machine (LightGBM) and long short-term memory (LSTM) algorithms to implement the proposed approach, and the framework is general and not limited to the algorithms selected in this study. Finally, the proposed approach is compared with other methods on eight real-life event logs. The experimental results show that the proposed approach can select effective features and improve prediction accuracy. In addition, the prediction results are explained.

**Key words:** process mining; remaining time prediction; feature selection; explainability; hierarchical model

随着企业数字化转型和智能化发展,企业信息系统中的事件数据得到了高质量的储存.流程挖掘技术可以从历史事件日志中提取有价值的信息<sup>[1-3]</sup>,帮助企业提高生产效率和产品质量,避免不必要的风险,提高企业的竞争力.经典的流程挖掘技术是指从现有事件日志中挖掘知识以发现、监控和改进实际流程<sup>[4,5]</sup>,建立起了数据挖掘与业务流程管理之间的联系,为业务流程管理提供了更加系统、高效的思路和技术.这些方法主要是通过已有的历史数据分析业务流程,但在业务的执行过程中,对流程未来执行情况的了解更有利于风险的提前掌握、早期预备和有效防范.因此,流程预测性监控是当前流程挖掘领域中的一个研究热点<sup>[6]</sup>.

流程预测性监控中常见的预测任务包括剩余时间预测、结果预测、下一事件预测、下一事件执行时间预测、后缀预测以及其他属性预测等<sup>[7,8]</sup>.剩余时间预测任务可以对正在运行实例的未来执行时间做出预判,帮助用户尽早调整后续的执行步骤和时间,避免因超时而带来的风险.目前对于业务流程剩余时间预测的研究中存在两个局限:(1)大多数现有剩余时间预测方法采用活动及从时间戳中提取的时间信息作为输入特征,或根据领域知识人为选择特征作为模型输入,缺少一个有说服力、较为通用的特征选择方法;(2)现有基于深度学习的预测模型难以解释输入特征和预测结果的内在联系,导致了不可解释的问题,用户难以信任并基于此做出相应决策.

为解决上述局限,本文提出了一个可解释特征分层框架预测流程剩余时间,主要贡献为:(1)提出一种通用的特征自选策略,包括基于优先级的后向特征删除和基于特征重要性值的前向特征选择两部分操作,在保证准确率的前提下减少选取特征的数量,降低后续构建模型的复杂程度;(2)提出一种可解释的特征分层输入预测模型结构,每层接收不同的特征,并将训练好的模型逐层输出进行可视化,以解释特征与预测结果的关联关系.通过 LightGBM (light gradient boosting machine) 和 LSTM (long short-term memory) 算法实现所提框架,并可选用其他方法进行替换.本文基于 8 个真实事件日志对所提方法进行实验对比和验证,选用其中一个日志进行案例分析,详述了特征选择和预测的整个过程.

本文第 1 节介绍相关工作.第 2 节介绍基础知识.第 3 节介绍可解释特征分层的流程剩余时间预测框架.第 4 节展示实验结果和案例分析.第 5 节对全文进行总结和展望.

## 1 相关工作

流程预测性监控是针对业务执行过程中未来执行步骤、时间、业务结果及执行资源等各种属性的预测<sup>[9-11]</sup>,通过对历史事件发生规律的探索,预测正在发生业务流程的未来情况.流程预测性监控包括基于流程感知和非流程感知两大类方法<sup>[12,13]</sup>.

基于流程感知的方法是借助流程发现从事件日志中挖掘模型的基础上进行预测<sup>[14,15]</sup>,主要结合发现模型和对历史事件的统计分布.van der Aalst 等人<sup>[16]</sup>提出一种带标记变迁系统作为预测模型的剩余时间预测方法,并研发支持工具 FSM Analyzer. Rogge-Solti 等人<sup>[17]</sup>采用一种随机 Petri 网 GDT\_SPN (stochastic Petri net with generally distributed transitions) 作为业务流程预测模型,用于预测流程剩余时间.流程感知方法通过显示模型进行预测,相对容易理解和解释,但所发现的模型质量依赖于不同的挖掘算法,因此预测结果不一定准确.

非流程感知依赖机器学习方法进行预测,随着事件日志中数据质量和数量的提升,深度学习模型也得到了广泛的应用. Tax 等人<sup>[18]</sup>将 LSTM 神经网络应用于业务流程预测中,并将活动和时间属性作为输入特征,相比流程感知方法,深度学习技术显著地提升了预测的准确率. Bukhsh 等人<sup>[19]</sup>将 Transformer 模型应用至剩余时间预测任务中,仅采用活动作为输入,显示了先进模型的优越性. Ni 等人<sup>[20]</sup>采用了 encoder-decoder 模型编码活动特征,获取了有效的上下文关系. Camargo 等人<sup>[21]</sup>采用 LSTM 神经网络构建预测模型,将资源属性加入到输入特征中并采用了词向量的编码方式. Pegoraro 等人<sup>[22]</sup>将文本属性经过编码作为输入特征之一,进一步提高了预测的准确率.可以看出,除了不同的编码方式,特征选取也是影响预测模型准确率的重要因素.

特征选择是机器学习方法中的数据预处理步骤,可以对原始特征进行降维,降低模型的复杂性,消除干扰特征,是提高算法性能的重要手段<sup>[23]</sup>. 特征选择主要包含过滤式、包裹式和嵌入式<sup>[24]</sup>. 许行等人<sup>[25]</sup>采用小样本数据,提出一种基于互信息的过滤式特征选择方法来降低数据的维度,但过滤式通常默认特征间相互独立,因此在一些情况下计算出来的相似度并不准确,进而影响学习模型的准确率. 黄南天等人<sup>[26]</sup>提出分类回归树的特征选择与最优决策树构建方法,采用嵌入式特征选择方法,获取特征重要性排序,确定最优特征子集,并进行剪枝操作获得最优分类树. 李占山等人<sup>[27]</sup>提出一种基于 LightGBM 的新型包裹式特征选择算法,采用 LightGBM 对原始特征构建迭代提升树模型并对特征重要性进行度量,结合前向搜索策略对特征进行选择,能够同时兼顾特征子集的计算效率和学习精度.

相比于流程感知和传统机器学习方法来说,虽然深度学习等技术不断被提出和应用,显著提升了预测的准确率,但可解释性不强难以理解,企业不能完全信任并基于此做出决策. Sindhgatta 等人<sup>[28]</sup>将机器学习可解释技术 LIME 应用至 XGBoost 算法构建的预测模型中,对基于 XGBoost 算法的预测结果提供了一定的后验解释. Harl 等人<sup>[29]</sup>将 GGNN 应用于业务流程预测中,通过可视化案例执行过程中不同活动对预测的影响程度,使预测具有可解释性. Hsieh 等人<sup>[30]</sup>设计了一个扩展的 DiCE 反事实方法,支持在轨迹的不同阶段推导具有里程碑感知的反事实条件,以促进可解释性. 当前的可解释工作主要是通过后验解释和流程可视化的方法来解释不同活动和输入特征对预测结果的影响,但难以在轨迹级别上解释各特征值与预测结果的关系.

综上,本文针对预测模型输入特征的选择问题和预测模型可解释的问题进行探索,尝试找到一种通用的包裹式特征选择方法,并为多特征输入的业务流程剩余时间预测任务构建可解释的预测模型.

## 2 基础知识

事件日志是一组轨迹的集合,每条轨迹都表示一次业务流程的执行. 每条轨迹由一系列事件组成,每个事件都表示一个活动的执行. 每个活动表示业务流程中的一个步骤.

**定义 1 (事件, 属性).** 设  $E$  为事件空间,即所有可能的事件标识符的集合,事件可以由不同的属性来描述. 设  $AN$  为属性名的集合,对于任意事件  $e \in E$  和属性名  $n \in AN$ :  $\#_n(e)$  是事件  $e$  的属性  $n$  的值.

设  $U_A$  为活动空间,  $\#_{act}(e) \in U_A$  表示事件  $e$  相关联的活动.

**定义 2 (轨迹, 案例和事件日志).** 一条轨迹  $\delta \in E^*$  是一个有限活动序列,其中每个事件只出现一次. 设  $C$  为案例空间,案例同事件一样也包含属性,对于任意  $c \in C$  和属性名  $n \in AN$ :  $\#_n(c)$  是案例  $c$  的属性  $n$  的值,每个案例包含一个特殊的属性,即轨迹  $\#_{trace}(c) \in E^*$ . 一个事件日志  $L \subseteq C$  是一组轨迹的集合,其中每个事件在整个日志中仅出现一次.

**定义 3 (轨迹前缀).**  $tp^k(\delta)$  是轨迹  $\delta$  的轨迹前缀,是由  $\delta$  的前  $k$  个元素组成的序列.

表 1 展示了 BPIC2012 日志的部分内容,仅显示了两条轨迹,包含 16 个事件,每个事件有唯一标识 id 和一些属性. 例如  $e_4$  活动 A\_PREACCEPTED 的实例,它发生于 2011/10/1 15:46,由编号 112 人员执行,其贷款金额为 13500. 案例 173703 的轨迹  $\delta = (e_1, e_2, e_3, e_4, e_5, e_6, e_7)$ ,  $k = 3$  时的轨迹前缀为  $tp^3(\delta) = (e_1, e_2, e_3)$ .

业务流程剩余时间预测的含义、目的和意义为:“我的业务还需要多长时间才能完成?”,为确保业务在需要的时间内完成办理,可根据预测结果进行调整,若出现延期情况可及时采取措施,不影响后续进展. 剩余时间预测任务的输入是已发生的事件序列,输出是从当前时刻到业务完成的预测时长. 一般来说用户可处于业务流程的任意一个事件节点执行预测任务.

表 1 BPIC2012 部分事件日志

Case id	Event id	Start time	Complete time	Activity	AMOUNT_REQ	Resource
173703	$e_1$	2011/10/1 15:45	2011/10/1 15:45	A_SUBMITTED	13500	112
	$e_2$	2011/10/1 15:45	2011/10/1 15:45	A_PARTLYSUBMITTED	13500	112
	$e_3$	2011/10/1 15:46	2011/10/1 15:46	A_PREACCEPTED	13500	112
	$e_4$	2011/10/1 15:46	2011/10/1 15:46	A_PREACCEPTED	13500	112
	$e_5$	2011/10/1 17:37	2011/10/1 17:40	W_Completeren aanvraag	13500	10912
	$e_6$	2011/10/1 18:40	2011/10/1 19:02	W_Completeren aanvraag	13500	10912
	$e_7$	2011/10/1 19:02	2011/10/1 19:02	A_CANCELLED	13500	10912
173709	$e_8$	2011/10/1 15:57	2011/10/1 15:57	A_SUBMITTED	11000	112
	$e_9$	2011/10/1 15:57	2011/10/1 15:57	A_PARTLYSUBMITTED	11000	112
	$e_{10}$	2011/10/1 15:58	2011/10/1 15:58	A_PREACCEPTED	11000	112
	$e_{11}$	2011/10/1 15:58	2011/10/1 15:58	A_PREACCEPTED	11000	112
	$e_{12}$	2011/10/1 16:26	2011/10/1 16:27	W_Completeren aanvraag	11000	10912
	$e_{13}$	2011/10/1 17:40	2011/10/1 17:43	W_Completeren aanvraag	11000	10912
	$e_{14}$	2011/10/10 23:42	2011/10/10 23:43	W_Completeren aanvraag	11000	10982
	$e_{15}$	2011/11/1 16:15	2011/11/1 16:15	W_Completeren aanvraag	11000	10982
	$e_{16}$	2011/11/1 16:15	2011/11/1 16:15	A_CANCELLED	11000	112
...	...	...	...	...	...	...

### 3 可解释特征分层的流程剩余时间预测框架

本节介绍可解释特征分层的流程剩余时间预测框架,其方法概述如图 1 所示,共分为以下 3 个阶段.

- (1) 特征选择: 将事件日志作为输入,自动选取对剩余时间预测任务积极有效的特征.
- (2) 特征编码: 将 (1) 中的已选特征作为输入,针对活动特征选用 Word2Vec 的编码方式,针对其他分类特征根据分类数的不同,选用随机向量或基于索引的编码方式.
- (3) 构建、训练可解释特征分层模型: 将编码后的特征按重要性排序依次作为模型每一层的输入,自动构建可解释特征分层模型,训练预测模型.

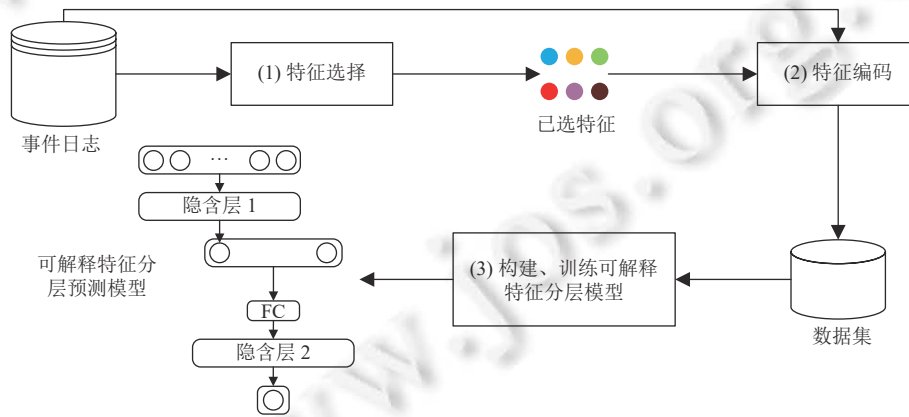


图 1 可解释特征分层的剩余时间预测框架概述

#### 3.1 特征自选取策略

特征选择的目的是在日志中选取对预测任务有积极影响的属性作为模型输入特征,这一过程不仅能够去除具有消极影响的属性,还可以减少输入特征保证模型的简洁性.由于不同特征对预测结果具有不同程度的影响,所以

需要判断是否将某特征保留或删除. 另外, 不同的特征组合可能也会影响到预测结果的准确性. 因此, 需要提供一种计算特征重要性的方法和特征自选取策略. 为判断特征自选取策略的有效性, 本文采用平均绝对误差 (mean absolute error,  $MAE$ ) 对剩余时间预测任务进行评价,  $MAE$  值是回归模型常用的评价指标, 计算真实值与预测值的平均绝对误差,  $MAE$  值越小说明预测的准确率越高, 其计算公式如下:

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (1)$$

其中,  $y$  为真实值,  $\hat{y}$  为预测值.

每次删除或增加某特征后, 需通过训练预测模型得到  $MAE$  值来判断某特征对预测的影响程度. 决策树是通过计算各节点的信息增益进行节点分裂, 因此特征的信息增益越高, 说明该特征在进行决策时越重要信息. 信息增益表示了各特征的重要程度, 为特征选择提供了初始参考指标. LightGBM 是一种高效的梯度提升决策树算法框架, 它采用了基于梯度的单侧抽样和互斥特征捆绑方法, 有效提高了算法的训练速度同时保证了算法的预测精度. 因此, 本文选用 LightGBM 实现特征选择阶段的预测模型<sup>[31]</sup>.

本文提出了一种通用的特征自选取策略, 共包含两个部分.

第 1 部分是基于优先级的后向特征删除策略, 主要用于筛除对预测任务具有消极影响的特征 (即增加该特征后降低了预测准确率). 考虑到两个或以上的某些捆绑特征可能共同对预测结果产生积极影响 (即增加该特征后提高了预测准确率), 而其中一个特征单独加入会造成消极影响, 故采取后向的删除策略, 以避免破坏特征间的组合关系. 然而, 当特征集中包含两个或以上的消极特征时, 删除一个消极特征可能会造成暂时的积极影响, 因此需要进行多轮逐步特征选择. 为特征设置优先级用于控制后向特征删除的轮数, 并且保证活动这一关键标志性特征不被删除.

在筛除过程中计算各特征删除前后的  $MAE$  差值, 作为新的特征重要性评判标准, 得到重要性值排序. 将决策树算法得到的特征重要性值集合  $I_{dt}$  作为初始参考标准, 将全部属性作为初始已选特征集合  $F = \{f_1, \dots, f_n\}$ , 在迭代筛除过程中计算  $f_i \in F$  删除前后的  $MAE$  差值, 作为特征重要性评判标准, 记为  $I_{MAE}(f_i)$ , 计算公式如下:

$$I_{MAE}(f_i) = MAE(F) - MAE(F - \{f_i\}) \quad (2)$$

其中,  $MAE(F)$  表示采用  $F$  所得的平均绝对误差,  $MAE(F - \{f_i\})$  表示采用删除  $f_i$  后的  $F$  集合得到的平均绝对误差. 基于优先级的后向特征删除策略详述步骤如下.

Step 1. 设置活动为最高优先级, 其他特征为 0, 采用已选特征集合  $F$ , 训练预测模型得到  $I_{dt}$ .

Step 2. 删除  $F$  中优先级最低且在  $I_{dt}$  集中值最低的特征  $f_i$ .

Step 3. 将删除特征  $f_i$  后的  $F$  重新训练预测模型, 计算预测模型在测试集上的  $MAE$  值, 得到  $I_{MAE}(f_i)$ , 若  $I_{MAE}(f_i) < 0$  说明  $f_i$  为消极特征, 否则增加其优先级并撤回删除操作, 若  $I_{MAE}$  集合中的所有值均大于 0 说明  $F$  中无消极特征, 执行 Step 4, 否则迭代执行 Step 2.

Step 4. 返回预测结果最佳的特征组合  $F$  和各特征的重要性值集合  $I_{MAE}$ .

第 2 部分是基于特征重要性值的前向特征选择策略. 由于第 1 部分避免破坏特征组合导致选取出的特征数量过多, 影响后续预测模型的简洁性和训练效率, 且许多特征仅对预测结果有轻微影响. 因此根据第 1 部分重要性值排序前向选择特征, 以达到在所选特征数尽可能少的情况下, 保证预测的准确率. 第 1 部分得到的  $F$  作为待选特征集合, 设置新的已选特征集合  $NF$ , 初始仅包含活动. 不同特征组合作为模型输入训练、测试所得的  $MAE$  值记录在集合  $FC_{MAE}$  中,  $FC_{MAE}(NF)$  表示特征组合  $NF$  对应的  $MAE$  值. 基于特征重要性值的前向特征选择策略的具体步骤如下.

Step 1. 采用  $NF$  作为输入重新训练预测模型, 计算预测模型在测试集上的  $MAE$  值.

Step 2. 若  $F$  为空执行 Step 3, 否则从  $F$  中选取  $I_{MAE}$  最大的特征  $f_i$  移至  $NF$ , 采用  $NF$  作为输入重新训练预测模型, 计算预测模型在测试集上的  $MAE$  值, 保存至  $FC_{MAE}(NF)$ , 迭代执行 Step 2.

Step 3. 返回  $FC_{MAE}$  中预测结果最佳的特征组合和  $MAE$  值.

### 3.2 特征编码

特征编码首先要将属性值转化为模型可计算的数据类型,其次编码应能高质量地表示上下文之间的内在关系.针对分类特征,简单的基于索引的编码和 one-hot 编码方式都存在一些弊端.基于索引的编码方式仅由一位整数组成,难以有效表示特征值并且不利于模型参数的训练;one-hot 编码方式是由 0、1 组成的向量,向量的位数即为特征的分类数,当表示某一特征值时,表示该特征值的位上的值为 1 其余均为 0. One-hot 编码虽能有效区分和代表不同类别,但本质上是一个稀疏矩阵,不能表示各类间的关系.而且,随着分类数的增加位数会不断增加,影响预测模型的训练效果.因此需要更高质量的编码方式来表示特征.

编码方式主要来自自然语言处理领域的词嵌入技术,包括 Word2Vec、FastText、LSA、GloVe、ELMO、GPT 和 BERT 等<sup>[32-34]</sup>.其中 Word2Vec 和 FastText 基于局部语料,优化效率高;LSA 利用全局语料特征,但求解计算复杂度大;GloVe 基于全局语料,结合了 LSA 和 Word2Vec 的优点;ELMO、GPT 和 BERT 采用了动态特征,可以解决一词多意的问题.在业务流程的预测性监控之中,特征值的分类数不会像自然语言的词汇量那样庞大,Word2Vec 即可满足特征值的表示. Word2Vec 包括 CBOW 和 Skip-gram 两种训练方式, CBOW 是采用上下文语料训练当前词, Skip-gram 是采用当前词训练上下文.

在本文的研究场景中,根据先前发生的事件及其属性进行业务流程剩余时间的预测.结合 CBOW 词向量训练方法,本文通过先前发生事件中的活动序列训练预测下一事件的活动,以得到活动的特征编码;其他分类特征难以判断其上下文关系,则采用随机初始化向量编码的方式,若特征值的类别小于 5 则采用基于索引的编码方式.

活动的编码原理如图 2 所示,输入层为最近的  $n$  个事件活动的 one-hot 向量表示,例  $n=3$ ,近 3 个事件为  $a_1, a_2, a_4$ ,则 one-hot 向量表示为  $[1, 1, 0, 1, \dots, 0, 0]$ ,隐藏层的单元个数即为向量编码的维数,输出层为下一事件的 one-hot 向量表示.通过计算预测的准确率调整隐含层单元个数,使之调整到最佳表示的向量维数,预测的准确率越高说明该向量表示事件的能力越强,最后将输入层与隐含层之间的参数权重作为词向量矩阵,每行表示一个活动的向量.

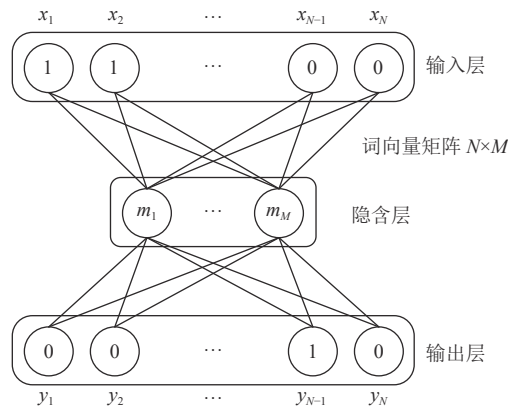


图 2 活动编码原理示意图

### 3.3 可解释特征分层模型

神经网络广泛应用于下一事件和剩余时间等预测任务中,由各层神经单元的权重、偏置以及激活函数组成,本质上可以通过各节点的权重解释特征对预测结果的影响,但当构建的神经网络结构复杂、参数繁多时,难以说明输入特征与预测结果的关联关系.本文提出一种特征分层的模型结构,将第 3.1 节得到的最优特征组合按照重要性值顺序输入至模型中,并将每层的输出进行可视化展现,以解释各特征对预测结果的影响.

特征分层输入的模型结构示例如图 3 所示,若所选特征按重要性值排序为活动、执行时间和资源,则依顺序

构造分层模型,在输入层1中只将活动经第3.2节得到的表示向量作为输入,经隐含层1后得到输出层1预测的剩余时间;在输入层2中将输出层1和当前事件的执行时间共同作为输入,由于执行时间是数值特征,则先经过一个全连接层(full connection, FC)以获取丰富的信息再输入至LSTM层中,经隐含层2后得到输出层2预测的剩余时间;在输入层3中将输出层2和当前事件的资源共同作为输入,资源作为分类特征输入的是其向量表示,经隐含层3后得到输出层3为最终预测的剩余时间。

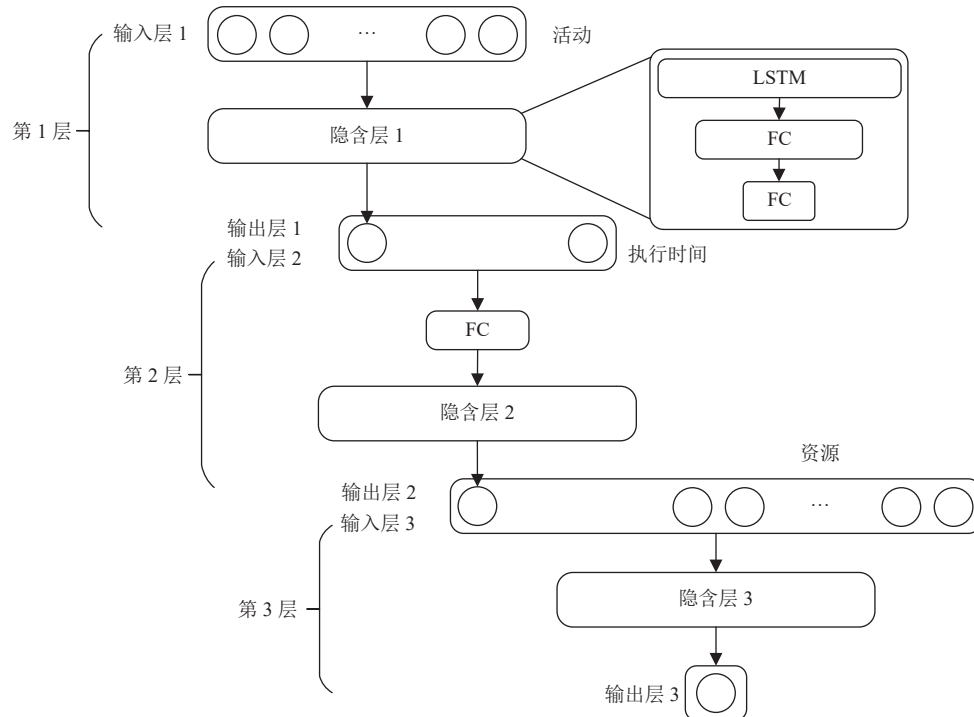


图3 可解释特征分层模型的结构示例

每层的输出是采用当前已输入的特征得到的剩余时间预测结果,可将每层结果可视化到二维平面,观察每加入一层特征对预测结果的影响。

需要说明模型结构中的隐含层可采用任意的网络结构进行替换,本文通过长短期记忆网络LSTM和全连接网络进行实现。LSTM是一种特殊的循环神经网络(recurrent neural network, RNN)<sup>[35,36]</sup>,是RNN的一种变体,RNN具有记忆功能是由于其独特的结构和工作原理,当前时刻的输出是由当前时刻的输入和上一时刻的状态共同决定的,因此能够对前面数据产生记忆。LSTM能够解决RNN存在的长期依赖问题,善于处理和预测时间序列中间隔较长的重要事件,具有长期记忆功能。适合流程预测中序列结构的应用,也是目前流程预测中应用最多的神经网络单元结构。我们可将预测节点的整个轨迹前缀作为模型输入,以获取更丰富的信息。

LSTM具有长期记忆取决于它的单元结构,如图4所示,每一个时序的特征输入都经过这些单元,单元结构中引入了门概念,通过控制遗忘门、输入门和输出门的状态信息计算出当前时刻的预测值。其中 $t$ 表示当前时刻, $x$ 表示输入, $h$ 表示输出, $C$ 表示单元状态, $\sigma$ 表示Sigmoid层,tanh表示双曲正切层。

预测值的计算过程如下。

第1步,由遗忘门的Sigmoid层决定从上一单元状态中遗忘多少信息, $h_{t-1}$ 和 $x_t$ 通过Sigmoid激活函数得到的 $f_t$ 为0到1之间的一个数字,1代表完全保留信息,0代表完全丢弃信息, $f_t$ 的计算公式为:

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fx}x_t + b_f) \quad (3)$$

第 2 步, 由输入门的 Sigmoid 层决定当前时刻的输入有哪些保存到单元状态, 计算  $i_t$ , 由 tanh 层计算当前状态候选值  $\tilde{C}_t$ , 然后将遗忘门的结果乘以上一时刻状态再加上输入门的结果乘以当前状态候选值得到当前时刻的状态信息  $C_t$ , 计算公式如下:

$$i_t = \sigma(W_{ih}h_{t-1} + W_{ix}x_t + b_i) \quad (4)$$

$$\tilde{C}_t = \tanh(W_{Ch}h_{t-1} + W_{Cx}x_t + b_C) \quad (5)$$

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t \quad (6)$$

第 3 步, 由输出门的 Sigmoid 层决定要输出哪些单元状态, 计算  $o_t$ , 将单元状态通过 tanh 函数归一化至  $[-1, 1]$ , 然后与输出门的结果相乘, 得到当前时刻的输出值  $h_t$ , 计算公式如下:

$$o_t = \sigma(W_{oh}h_{t-1} + W_{ox}x_t + b_o) \quad (7)$$

$$h_t = o_t \tanh(C_t) \quad (8)$$

以上可得出在  $t$  时刻的预测值  $h_t$ , 公式 (3)–(8) 中, 权重矩阵  $W_{fh}$ 、 $W_{ih}$ 、 $W_{Ch}$ 、 $W_{oh}$ 、 $W_{fx}$ 、 $W_{ix}$ 、 $W_{Cx}$ 、 $W_{ox}$  和偏置  $b_f$ 、 $b_i$ 、 $b_C$ 、 $b_o$  为 LSTM 需训练的 12 组参数。

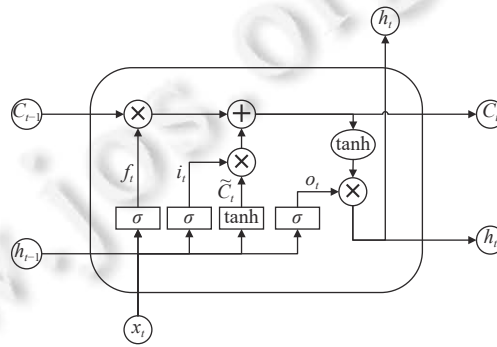


图 4 LSTM 单元结构

在模型的训练过程中, 每一个批次都逐层训练, 即先训练第 1 层, 计算输出层 1 的预测值与真实值的 loss, 进行反向传播调整网络结构参数, 然后固定第 1 层的参数, 训练第 2 层, 计算输出层 2 的 loss, 以此类推, 直至每一层都得到充分的训练。

## 4 实验设置和结果分析

本节介绍了测试本文方法的实验设置和所用数据集, 对本文所提方法的实验过程和结果进行了展示和分析, 并将实验结果与其他方法进行比较, 以 Helpdesk 事件日志为例进行模型预测的可视化展现. 本文方法在 Python 3.7 环境下实现, 所使用的计算机的配置为: Windows 11 操作系统, AMD Ryzen 7 5800H with Radeon Graphics @ 3.20 GHz 处理器和 16.00 GB 的内存, 代码开源到 GitHub (<https://github.com/gn874682003/Hierarchical-Prediction-Framework>).

### 4.1 实验设置

(1) 数据集划分: 本文为每个事件日志训练一个预测模型. 训练集和测试集根据 8:2 的比例进行划分, 将事件日志按时间划分为 5 部分, 每一部分选取 1/5 的轨迹作为测试集, 其余轨迹为训练集.

(2) 特征自选取: 在特征选取阶段, 除了现有属性还通过时间戳拓展出 6 个数值属性, 分别为执行时间 duration、总执行时间 allDuration、月份 month、日期 day、星期 week 和小时 hour. 其中 duration 是指上一个事件结束到当前事件结束的时长, allDuration 是指从该案例开始到当前事件结束的时长. LightGBM 算法需设置轨迹前缀长度为 1 的数据集训练和测试预测模型.



(3) 特征编码: 词向量的训练需设置固定长度的上文, 本文从前缀长度分别为 1、3、5 的设置中选择在 CBOW 训练阶段预测下一事件准确率最高的设置, 从而确定特征编码阶段的前缀长度为 3。

(4) 可解释特征分层预测模型: LSTM 可将所有轨迹前缀信息循环输入至网络单元, 故选取可变长的前缀。适合循环神经网络的可变前缀长度有两种生成方式, 例如一条事件的执行轨迹为  $[e_1, e_2, e_4, e_3, e_6]$ 。第 1 种是以整条轨迹作为一组数据, 即输入为  $[e_1, e_2, e_4, e_3, e_6]$ , 输出为每个事件的真实剩余时间序列  $[4.2, 3.1, 2.9, 1.2, 0]$ 。第 2 种是切分轨迹划分数据, 可得到 5 组数据, 输入包含  $[e_1]$ ,  $[e_1, e_2]$ ,  $[e_1, e_2, e_4]$ ,  $[e_1, e_2, e_4, e_3]$ ,  $[e_1, e_2, e_4, e_3, e_6]$ , 输出则分别对应  $[4.2]$ ,  $[3.1]$ ,  $[2.9]$ ,  $[1.2]$ ,  $[0]$ 。两种划分方法均可实现在轨迹的任一事件节点执行预测任务, 在事件日志轨迹数量较少的情况下可采用第 2 种划分方式, 增加训练集的规模, 能够更充分地训练预测模型。本文经过实验应用神经网络构建预测模型的超参数设置如下: 学习率为 0.001, 优化算法选用 Adam, 批处理大小为 100, 迭代次数为 300, 隐含层节点数以及向量维数的设置针对不同的事件日志进行调整。

## 4.2 数据集

本文实验使用 8 个来自 4TU Center for Research 的公开事件日志数据集, 来验证本文所提方法。Helpdesk 事件日志涉及一家意大利软件公司帮助台的票务管理流程 (<https://doi.org/10.4121/uuid:0c60edf1-6f83-4e75-9367-4c63b3e9d5bb>); BPIC2012 来自 2012 年 BPI 挑战赛, 是某财政机构贷款申请流程的事件日志 (<https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f>); BPIC2015 来自 2015 年 BPI 挑战赛, 共包含 5 个事件日志, 分别由 5 个荷兰市政府提供, 数据包含大约 4 年期间所有建筑许可证申请 (<https://doi.org/10.4121/uuid:31a308ef-c844-48da-948c-305d167a0ec1>); Production 事件日志来自某生产车间 2012 年 1-3 月部分产品的生产流程数据 (<https://doi.org/10.4121/uuid:68726926-5ac5-4fab-b873-ee76ea412399>)。

日志的基本统计属性包括活动个数、轨迹长度、变体数量等, 表 2 给出了 8 个事件日志的统计属性。对于数据量较小的 Production 日志采用可变前缀长度的第 2 种切分轨迹划分数据的方式生成训练集和测试集, 该数据集经过实验设置前缀长度范围的取值为  $[1, 10]$ , 因为过长的轨迹前缀在迭代过程中会遗忘早期信息, 对准确率的提升不明显, 还会影响训练效率, 第 1 种划分方式采用了全部前缀, 因此无需设置前缀长度范围。

表 2 事件日志的统计属性

数据集	轨迹数	事件数	活动数	属性数	变体数	事件数 (每条轨迹)		
						最小值	平均值	最大值
Helpdesk	4580	21348	14	15	226	2	5	15
BPIC2012	13087	262200	36	6	4366	3	20	175
BPIC2015_1	1199	52217	398	28	1170	2	44	101
BPIC2015_2	832	44354	410	27	828	1	53	132
BPIC2015_3	1409	59681	383	28	1349	3	42	124
BPIC2015_4	1053	47293	356	28	1049	1	45	116
BPIC2015_5	1156	59083	389	28	1153	5	51	154
Production	225	4543	55	14	221	1	20	175

## 4.3 特征自选取策略的实验对比结果

为验证特征自选取策略所选特征对业务流程预测任务的有效性, 采用活动属性、全部属性以及本文方法所选属性分别作为输入特征来进行对比实验。实验的预测结果如表 3 所示, activity 表示仅选用“活动”作为输入特征, all 表示选取所有属性作为输入特征, FeaSelPart1 表示经过特征自选取策略第 1 部分所选属性作为输入特征, FeaSelPart2 表示经过整个策略所选属性作为输入特征。本文的 MAE 值以天为单位, 表示所有流程轨迹前缀的真实剩余时间与预测剩余时间的平均绝对误差。

由表 3 可以看出, 在 8 个事件日志中除 BPIC2015\_3 日志外, 其他日志选用 All 比单一使用 Activity 活动属性作为预测模型的输入特征会大幅提高预测的准确率, BPIC2015\_3 日志中可能存在某些属性对预测具有干扰作用。

通过 FeaSelPart1 筛除消极特征后, *MAE* 值在各事件日志中均有显著的降低. FeaSelPart2 在尽量保证预测准确率不大幅下降的情况下减少所选的特征数. 可以看出选取后的特征组合不仅在多数日志中 *MAE* 值有所下降, 还显著减少了所选特征数量, 为后续工作提供了简洁、有效的数据支持, 另外特征选择的执行时间较短, 可以保证执行效率.

表 3 不同特征组合在预测任务上的表现对比

数据集	Activity	All		FeaSelPart1			FeaSelPart2		
	<i>MAE</i> (天)	<i>MAE</i> (天)	特征数	<i>MAE</i> (天)	特征数	特征选择策略的执行时间 (s)	<i>MAE</i> (天)	特征数	特征选择策略的执行时间 (s)
Helpdesk	6.575	4.917	18	4.684	10	16	<b>4.659</b>	<b>8</b>	<b>1</b>
BPIC2012	7.483	7.207	9	7.192	8	17	<b>7.192</b>	<b>8</b>	<b>4</b>
BPIC2015_1	40.143	32.062	22	<b>26.364</b>	<b>14</b>	<b>38</b>	30.505	8	3
BPIC2015_2	80.278	63.224	22	62.405	18	53	<b>59.472</b>	<b>8</b>	<b>2</b>
BPIC2015_3	23.590	26.811	22	19.001	9	26	<b>18.594</b>	<b>5</b>	<b>3</b>
BPIC2015_4	63.098	53.445	22	49.451	13	41	<b>48.475</b>	<b>5</b>	<b>1</b>
BPIC2015_5	49.094	43.677	22	<b>37.613</b>	<b>15</b>	<b>44</b>	38.579	8	8
Production	18.312	12.919	16	12.550	11	11	<b>12.469</b>	<b>7</b>	<b>1</b>

#### 4.4 可解释特征分层模型的实验对比结果

为验证分层模型和编码方式对预测任务的影响, 设置了如表 4 所示的对比实验并展示了测试结果. InAct 表示采取基于索引的编码方式并仅使用活动作为输入特征, InEFH 表示采取基于索引的编码方式并使用 FeaSelPart2 所选属性作为输入特征, EembAct 表示采取第 3.2 节所述的编码方式并仅使用活动作为输入特征, EFH 为本文所提方法.

表 4 多属性、编码和分层在预测任务上的 *MAE* 表现对比 (天)

数据集	InAct	InEFH	EembAct	EFH
Helpdesk	6.587	6.411	4.582	<b>4.270</b>
BPIC2012	7.332	7.079	7.686	<b>6.986</b>
BPIC2015_1	32.186	28.125	26.998	<b>25.493</b>
BPIC2015_2	76.240	71.748	<b>64.132</b>	66.993
BPIC2015_3	21.564	18.862	17.993	<b>17.323</b>
BPIC2015_4	59.308	52.940	50.973	<b>50.227</b>
BPIC2015_5	40.397	33.605	35.278	<b>33.464</b>
Production	17.584	18.004	13.220	<b>11.510</b>

由表 4 可得以下结论, 在多数日志中 InEFH 的 *MAE* 值均小于 InAct, 说明本文策略所选特征能够提高预测的准确率. 在多数日志中 EembAct 的预测结果优于 InEFH 和 InAct, 说明本文应用的活动编码方式能够有效获取活动间的关系. EFH 在除 BPIC2015\_2 的日志上均取得了最好的结果, 说明分层模型的构造能够有效利用每一个特征值, 达到较好的结果. 然而, BPIC2015\_2 日志的 EFH 方法相较 EembAct 方法的 *MAE* 值有所升高, 并且表 3 中依托于 LightGBM 的 FeaSelPart2 在 BPIC2015\_2 和 BPIC2015\_4 日志中得到的 *MAE* 值比 EFH 更低, 这可能是由于某些特征组合需同时发挥作用, 而分散在不同的层次之间则产生消极影响. 本文还未考虑将具有组合作用的特征放入同一层内, 这可作为未来的研究方向.

表 5 给出了本文所提方法与其他方法的比较结果, 其中 LSTMNN<sup>[18]</sup> 是 Tax 等人提出的基于 LSTM 的业务过程预测模型, 采用了活动和时间属性作为输入特征, LSTM 是最早应用深度学习模型到流程预测中的方法. Process Transformer<sup>[19]</sup> 是 Bukhsh 等人提出的基于 Transformer 的业务流程预测模型, 该方法添加了时间信息作为输入, 是先进深度学习方法在流程预测领域中的应用. Auto-encoded<sup>[20]</sup> 是对活动前缀进行编码以获取丰富的上下文关系, 该方法只采用活动这一关键属性.

表5 可解释模型与其他算法的 MAE 表现对比 (天)

数据集	LSTMNN <sup>[18]</sup>	Process Transformer <sup>[19]</sup>	Auto-encoded <sup>[20]</sup>	EFH
Helpdesk	62.089	5.419	8.733	<b>4.270</b>
BPIC2012	61.175	<b>5.968</b>	8.4268	6.986
BPIC2015_1	49.648	39.472	38.324	<b>25.493</b>
BPIC2015_2	127.834	87.622	88.895	<b>66.993</b>
BPIC2015_3	61.190	17.544	17.588	<b>17.323</b>
BPIC2015_4	151.565	43.777	<b>39.619</b>	50.227
BPIC2015_5	120.750	51.847	38.270	<b>33.464</b>
Production	521.296	14.992	11.775	<b>11.510</b>

由表5可以看出,本文所提方法在多数事件日志中表现最好,这是由于选取了更多的有效属性,为预测模型提供了丰富的已知信息,提高了剩余时间预测结果的准确性.在BPIC2012日志中Process Transformer预测模型表现最好,这是由于该日志的规律性较强,受其他属性影响较少,并且数据规模较大能充分训练Transformer模型.在BPIC2015\_4日志中Auto-encoded预测表现最好,通过轨迹执行时间分析可得该日志在BPIC2015系列日志中异常值最少,受其他属性影响较少,通过自编码技术能够有效区分不同轨迹前缀,得到较好的结果.而在其他日志上两种方法的表现参差不齐,说明不同类型的事件日志适合不同的预测模型.本文添加了其他有效特征使得在LSTM这一较为基础的模型上也能表现良好,说明了特征选择和分层结构对于提升剩余时间预测准确率的有效性.本文提出的是一种通用的可解释特征分层预测框架,实现方法可使用其他算法进行替换,若采用拟合能力更强的预测模型可能会得到更好的预测结果.

#### 4.5 案例分析

本节以Helpdesk事件日志为例,展示了特征选择过程并可视化解了预测结果.图5是经过特征选取策略第1部分之后根据 $I_{MAE}$ 画出的特征重要性图,说明了各特征对预测结果的影响程度,提供了全局解释.图中只显示了对预测具有积极影响的属性,活动作为关键标志特征,不需要计算其特征重要性.

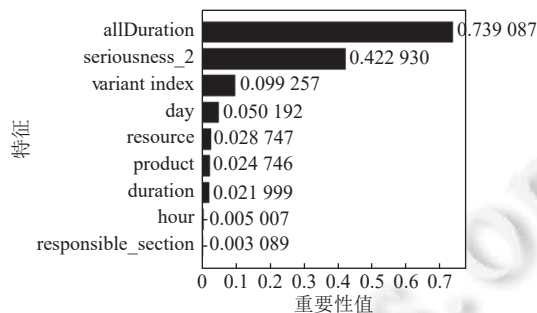


图5 Helpdesk 的特征重要性值

特征选择策略的第2部分筛选过程如表6所示,表6中No.0是策略第1部分选取的特征组合和预测结果, No.1-10展示了策略第2部分每次迭代使用的特征组合和预测结果.策略第2部分返回No.8特征组合为最终结果,相比于No.0去除了hour和responsible\_section得到了最小的MAE值,可以看出这两个属性单独去除都会增加预测误差,可两个属性同时去除则可以减少预测误差,说明特征间具有组合关系.

根据特征选择策略获得的特征组合No.8作为输入训练分层模型,模型根据特征重要性值的排序逐层构建网络,测试结果如图6所示.可以看出预测误差整体呈逐层下降的趋势,但在variant index和product的属性层上略有上升,最后一层得到误差最小值4.27.

图7随机抽取了测试集中来自4条轨迹中4个事件的预测结果,并可视化出了每层的预测值.可以看出allDuration这一属性对预测有较大的积极影响,它总能得到比仅用activity更接近真实值的结果,由图7(a)和图7(c)

可以看出 allDuration 较大时剩余时间会相对减少,所以它的预测值与第 1 层预测值相比会减少,而图 7(b)和图 7(d)中 allDuration 非常小则预测往往会偏大.通过图 7(c)和图 7(d)对比来看更能验证这一结论,图 7(c)和图 7(d)中的 activity 均为“take in charge ticket”,预测值相同,但真实值相差较大,通过不同的 allDuration 值得到了更接近真实值的预测结果.其他属性也具有不同程度的积极影响,每个属性根据当前特征值的不同改变预测结果和影响趋势(例如图中位于真实值的上下两侧).由于该真实日志的发布者对敏感隐私数据进行了脱敏处理,许多特征值由代号来进行表示.若在具体场景中知道 Value1 等代号所指的具体含义,则可对预测结果产生更丰富的解释.可以看出,我们的方法能够在事件层面上解释每一次预测的输入特征值对预测结果的影响.

表 6 Helpdesk 特征组合预测结果

No.	特征组合	MAE (天)	特征数
0	activity, allDuration, seriousness_2, variant index, day, resource, product, duration, hour, responsible_section	4.684	10
1	activity	6.575	1
2	activity, allDuration	5.223	2
3	activity, allDuration, seriousness_2	4.786	3
4	activity, allDuration, seriousness_2, variant index	4.744	4
5	activity, allDuration, seriousness_2, variant index, day	4.722	5
6	activity, allDuration, seriousness_2, variant index, day, resource	4.704	6
7	activity, allDuration, seriousness_2, variant index, day, resource, product	4.699	7
<b>8</b>	<b>activity, allDuration, seriousness_2, variant index, day, resource, product, duration</b>	<b>4.659</b>	<b>8</b>
9	activity, allDuration, seriousness_2, variant index, day, resource, product, duration, hour	4.687	9
10	activity, allDuration, seriousness_2, variant index, day, resource, product, duration, responsible_section	4.689	9

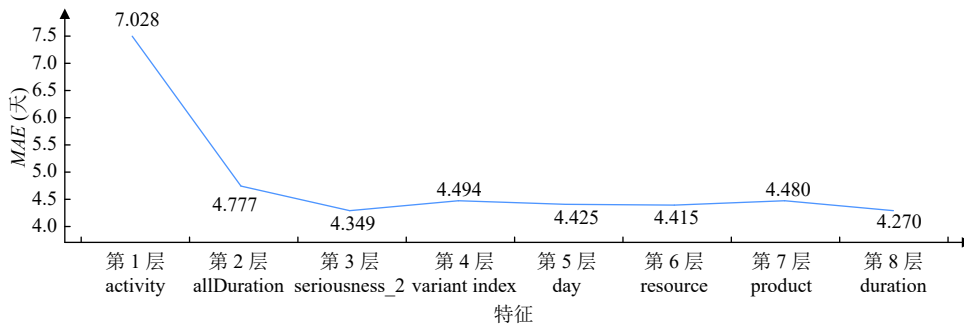
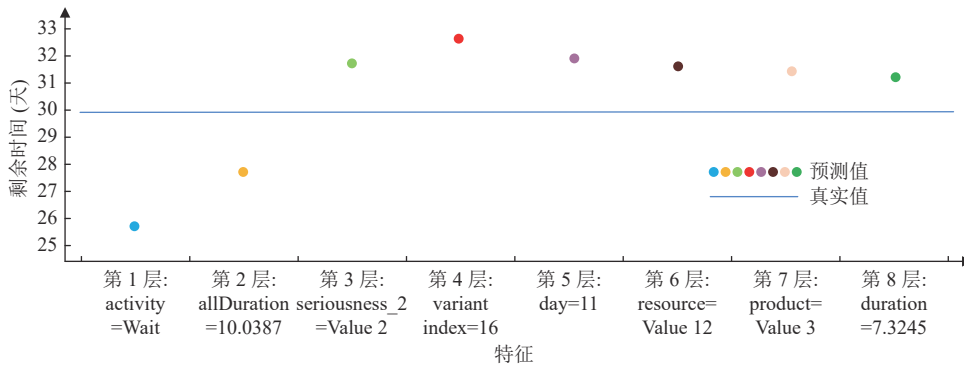
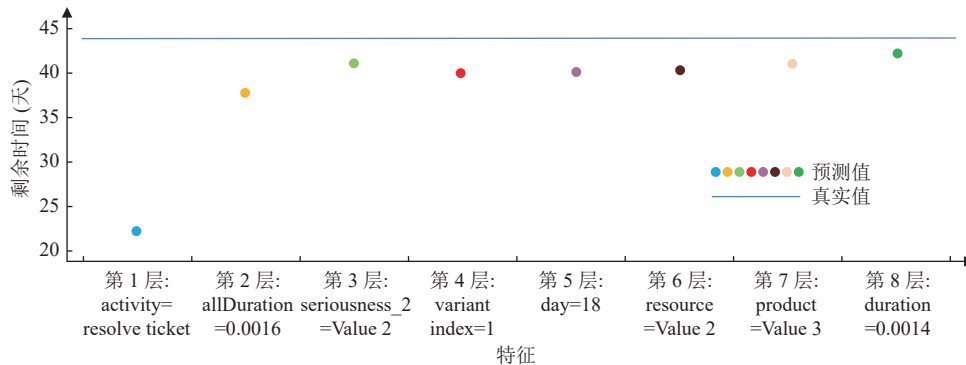


图 6 Helpdesk 分层模型的逐层预测结果

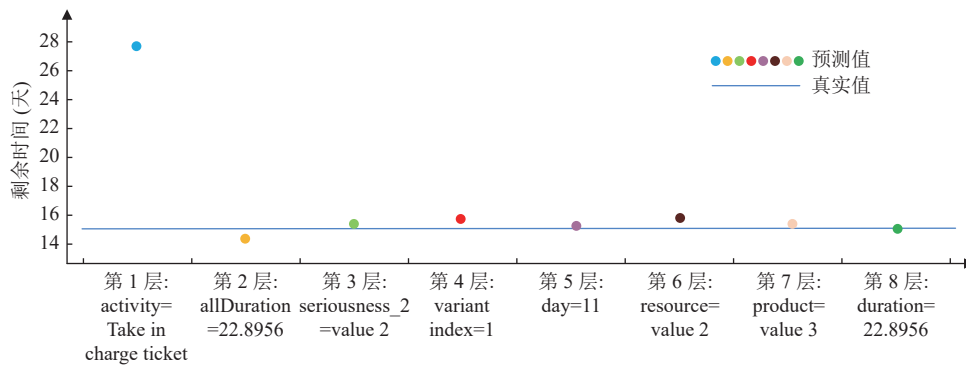


(a) “Case 1899” 第 3 个事件节点

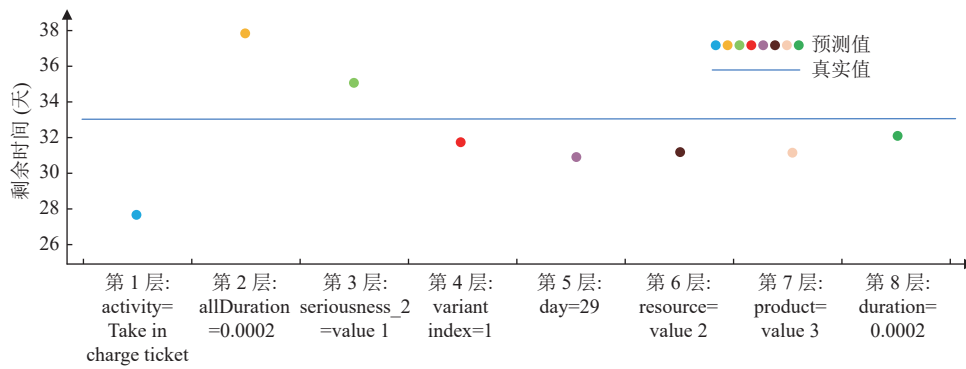
图 7 Helpdesk 日志预测结果示例



(b) “Case 390” 第3个事件节点



(c) “Case 404” 第2个事件节点



(d) “Case 4505” 第2个事件节点

图7 Helpdesk 日志预测结果示例 (续)

### 5 结论

基于深度学习的业务流程剩余时间预测方法表现出了较强的拟合与泛化能力,得到了更高的准确率. 现有的工作大多只采用活动特征或根据领域知识选择输入特征构建预测模型,缺少通用的特征选择方法. 另外基于深度学习的预测模型由于其结构的复杂性难以解释输入特征与预测结果的内在联系. 为解决上述问题,本文的主要贡

献为: (1) 提出一种通用的特征自选取策略, 在保证准确率的前提下减少选取特征的数量, 降低后续构建模型的复杂程度; (2) 提出一种可解释特征分层预测模型结构, 每层接收不同的特征, 并将训练好的模型逐层输出进行可视化, 以解释特征与预测结果的关联关系. 本文通过 LightGBM 和 LSTM 算法实现所提框架, 并基于 8 个真实事件日志对所提方法进行实验对比和验证, 选用其中一个日志进行案例分析, 详述了特征选择和预测的整个过程. 实验验证了特征自选取策略的有效性, 提供了一种可信的解释方法, 提升了剩余时间预测任务的准确率.

未来研究工作主要在所提框架中采用更先进的深度学习模型替换当前的 LSTM, 以验证是否可以进一步提升预测准确率. 其次, 尝试确定具有组合关系的特征, 将其放入同一层中以验证特征结合对预测结果的影响. 此外, 本文通过特征这一角度对预测的可解释性进行了探讨, 可以继续探讨其他可解释的模型或方法, 进一步提高模型的可信性.

## References:

- [1] van der Aalst WMP. *Process Mining: Discovery, Conformance and Enhancement of Business Processes*. Berlin: Springer, 2011. [doi: [10.1007/978-3-642-19345-3](https://doi.org/10.1007/978-3-642-19345-3)]
- [2] Liu C, Duan H, Zeng QT, Zhou MC, Lu FM, Cheng JJ. Towards comprehensive support for privacy preservation cross-organization business process mining. *IEEE Trans. on Services Computing*, 2019, 12(4): 639–653. [doi: [10.1109/TSC.2016.2617331](https://doi.org/10.1109/TSC.2016.2617331)]
- [3] van der Aalst W, Weijters T, Maruster L. Workflow mining: Discovering process models from event logs. *IEEE Trans. on Knowledge and Data Engineering*, 2004, 16(9): 1128–1142. [doi: [10.1109/TKDE.2004.47](https://doi.org/10.1109/TKDE.2004.47)]
- [4] Wen LJ, Wang JM, van der Aalst WMP, Huang BQ, Sun JG. A novel approach for process mining based on event types. *Journal of Intelligent Information Systems*, 2009, 32(2): 163–190. [doi: [10.1007/s10844-007-0052-1](https://doi.org/10.1007/s10844-007-0052-1)]
- [5] Buijs JCAM, van Dongen BF, van der Aalst WMP. A genetic algorithm for discovering process trees. In: *Proc. of the 2012 IEEE Congress on Evolutionary Computation*. Brisbane: IEEE, 2012. 1–8. [doi: [10.1109/CEC.2012.6256458](https://doi.org/10.1109/CEC.2012.6256458)]
- [6] Harane N, Rathi S. Comprehensive survey on deep learning approaches in predictive business process monitoring. In: Gunjan VK, Zurada JM, Raman B, Gangadharan GR, eds. *Modern Approaches in Machine Learning and Cognitive Science: A Walkthrough*. Cham: Springer, 2020. 115–128. [doi: [10.1007/978-3-030-38445-6\\_9](https://doi.org/10.1007/978-3-030-38445-6_9)]
- [7] Xu XR, Liu C, Li T, Guo N, Ren CG, Zeng QT. Business process remaining time prediction: An approach based on bidirectional quasi recurrent neural network with attention. *Acta Electronica Sinica*, 2022, 50(8): 1975–1984 (in Chinese with English abstract). [doi: [10.12263/DZXB.20211477](https://doi.org/10.12263/DZXB.20211477)]
- [8] Sun XX, Hou WJ, Ying YK, Yu DJ. Business process remaining time prediction based on two-layer machine learning. *Chinese Journal of Computers*, 2021, 44(11): 2283–2294 (in Chinese with English abstract). [doi: [10.11897/SP.J.1016.2021.02283](https://doi.org/10.11897/SP.J.1016.2021.02283)]
- [9] Márquez-Chamorro AE, Resinas M, Ruiz-Cortés A. Predictive monitoring of business processes: A survey. *IEEE Trans. on Services Computing*, 2018, 11(6): 962–977. [doi: [10.1109/TSC.2017.2772256](https://doi.org/10.1109/TSC.2017.2772256)]
- [10] Di Francescomarino C, Ghidini C, Maggi FM, Milani F. Predictive process monitoring methods: Which one suits me best? In: *Proc. of the 16th Int'l Conf. on Business Process Management*. Sydney: Springer, 2018. 462–479. [doi: [10.1007/978-3-319-98648-7\\_27](https://doi.org/10.1007/978-3-319-98648-7_27)]
- [11] Teinmaa I, Dumas M, La Rosa M, Maggi FM. Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Trans. on Knowledge Discovery from Data*, 2019, 13(2): 17. [doi: [10.1145/3301300](https://doi.org/10.1145/3301300)]
- [12] Verenich I, Dumas M, La Rosa M, Maggi FM, Teinmaa I. Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. *ACM Trans. on Intelligent Systems and Technology*, 2019, 10(4): 34. [doi: [10.1145/3331449](https://doi.org/10.1145/3331449)]
- [13] Neu DA, Lahann J, Fettke P. A systematic literature review on state-of-the-art deep learning methods for process prediction. *Artificial Intelligence Review*, 2022, 55(2): 801–827. [doi: [10.1007/s10462-021-09960-8](https://doi.org/10.1007/s10462-021-09960-8)]
- [14] Bolt A, Sepúlveda M. Process remaining time prediction using query catalogs. In: *Proc. of the 2013 Int'l Conf. on Business Process Management*. Beijing: Springer, 2013. 54–65. [doi: [10.1007/978-3-319-06257-0\\_5](https://doi.org/10.1007/978-3-319-06257-0_5)]
- [15] Pika A, van der Aalst WMP, Fidge CJ, Ter Hofstede AHM, Wynn MT. Profiling event logs to configure risk indicators for process delays. In: *Proc. of the 25th Int'l Conf. on Advanced Information Systems Engineering*. Valencia: Springer, 2013. 465–481. [doi: [10.1007/978-3-642-38709-8\\_30](https://doi.org/10.1007/978-3-642-38709-8_30)]
- [16] van der Aalst WMP, Schonenberg MH, Song M. Time prediction based on process mining. *Information Systems*, 2011, 36(2): 450–475. [doi: [10.1016/j.is.2010.09.001](https://doi.org/10.1016/j.is.2010.09.001)]
- [17] Rogge-Solti A, Weske M. Prediction of remaining service execution time using stochastic petri nets with arbitrary firing delays. In: *Proc. of the 11th Int'l Conf. on Service-oriented Computing*. Berlin: Springer, 2013. 389–403. [doi: [10.1007/978-3-642-45005-1\\_27](https://doi.org/10.1007/978-3-642-45005-1_27)]
- [18] Tax N, Verenich I, La Rosa M, Dumas M. Predictive business process monitoring with LSTM neural networks. In: *Proc. of the 29th Int'l*

- Conf. on Advanced Information Systems Engineering. Essen: Springer, 2017. 477–492. [doi: [10.1007/978-3-319-59536-8\\_30](https://doi.org/10.1007/978-3-319-59536-8_30)]
- [19] Bukhsh ZA, Saeed A, Dijkman RM. ProcessTransformer: Predictive business process monitoring with transformer network. arXiv:2104.00721, 2021.
- [20] Ni WJ, Yan M, Liu T, Zeng QT. Predicting remaining execution time of business process instances via auto-encoded transition system. *Intelligent Data Analysis*, 2022, 26(2): 543–562. [doi: [10.3233/IDA-215755](https://doi.org/10.3233/IDA-215755)]
- [21] Camargo M, Dumas M, González-Rojas O. Learning accurate LSTM models of business processes. In: Proc. of the 17th Int'l Conf. on Business Process Management. Vienna: Springer, 2019. 286–302. [doi: [10.1007/978-3-030-26619-6\\_19](https://doi.org/10.1007/978-3-030-26619-6_19)]
- [22] Pegoraro M, Uysal MS, Georgi DB, van der Aalst WMP. Text-aware predictive monitoring of business processes. In: Proc. of the 24th Int'l Conf. on Business Information Systems. Hannover: BIS, 2021. 221–232.
- [23] Agrawal P, Abutarboush HF, Ganesh T, Mohamed AW. Metaheuristic algorithms on feature selection: A survey of one decade of research (2009–2019). *IEEE Access*, 2021, 9: 26766–26791. [doi: [10.1109/ACCESS.2021.3056407](https://doi.org/10.1109/ACCESS.2021.3056407)]
- [24] Rostami M, Berahmand K, Nasiri E, Forouzandeh S. Review of swarm intelligence-based feature selection methods. *Engineering Applications of Artificial Intelligence*, 2021, 100: 104210. [doi: [10.1016/j.engappai.2021.104210](https://doi.org/10.1016/j.engappai.2021.104210)]
- [25] Xu X, Zhang K, Wang WJ. A feature selection method for small samples. *Journal of Computer Research and Development*, 2018, 55(10): 2321–2330 (in Chinese with English abstract). [doi: [10.7544/issn1000-1239.2018.20170748](https://doi.org/10.7544/issn1000-1239.2018.20170748)]
- [26] Huang NT, Peng H, Cai GW, Xu DG. Feature selection and optimal decision tree construction of complex power quality disturbances. *Proc. of the CSEE*, 2017, 37(3): 776–785 (in Chinese with English abstract). [doi: [10.13334/j.0258-8013.pcsee.160108](https://doi.org/10.13334/j.0258-8013.pcsee.160108)]
- [27] Li ZS, Yao X, Liu ZG, Zhang JC. Feature selection algorithm based on LightGBM. *Journal of Northeastern University (Natural Science)*, 2021, 42(12): 1688–1695 (in Chinese with English abstract). [doi: [10.12068/j.issn.1005-3026.2021.12.003](https://doi.org/10.12068/j.issn.1005-3026.2021.12.003)]
- [28] Sindhgatta R, Ouyang C, Moreira C, Liao Y. Interpreting predictive process monitoring benchmarks. arXiv:1912.10558, 2019.
- [29] Harl M, Weinzierl S, Stierle M, Matzner M. Explainable predictive business process monitoring using gated graph neural networks. *Journal of Decision Systems*, 2020, 29(S1): 312–327. [doi: [10.1080/12460125.2020.1780780](https://doi.org/10.1080/12460125.2020.1780780)]
- [30] Hsieh C, Moreira C, Ouyang C. DiCE4EL: Interpreting process predictions using a milestone-aware counterfactual approach. In: Proc. of the 3rd Int'l Conf. on Process Mining. Eindhoven: IEEE, 2021: 88–95. [doi: [10.1109/ICPM53251.2021.9576881](https://doi.org/10.1109/ICPM53251.2021.9576881)]
- [31] Ke GL, Meng Q, Finley T, Wang TF, Chen W, Ma WD, Ye QW, LiuTY. LightGBM: A highly efficient gradient boosting decision tree. In: Proc. of the 31st Int'l Conf. on Neural Information Processing Systems. Long Beach: ACM, 2017. 3149–3157. [doi: [10.5555/3294996.3295074](https://doi.org/10.5555/3294996.3295074)]
- [32] Mikolov T, Sutskever I, Chen K, *et al.* Distributed representations of words and phrases and their compositionality. In: Proc. of the 26th Int'l Conf. on Neural Information Processing Systems. Lake Tahoe: ACM, 2013. 3111–3119. [doi: [10.5555/2999792.2999959](https://doi.org/10.5555/2999792.2999959)]
- [33] Joulin A, Grave E, Bojanowski P, Douze M, Jégou H, Mikolov T. FastText.zip: Compressing text classification models. arXiv:1612.03651, 2016.
- [34] Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Minneapolis: ACL, 2018. 4171–4186. [doi: [10.18653/v1/N19-1423](https://doi.org/10.18653/v1/N19-1423)]
- [35] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*, 1997, 9(8): 1735–1780. [doi: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735)]
- [36] Sherstinsky A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, 2020, 404: 132306. [doi: [10.1016/j.physd.2019.132306](https://doi.org/10.1016/j.physd.2019.132306)]

#### 附中文参考文献:

- [7] 徐兴荣, 刘聪, 李婷, 郭娜, 任崇广, 曾庆田. 基于双向准循环神经网络和注意力机制的业务流程剩余时间预测方法. *电子学报*, 2022, 50(8): 1975–1984. [doi: [10.12263/DZXB.20211477](https://doi.org/10.12263/DZXB.20211477)]
- [8] 孙笑笑, 侯文杰, 应钰柯, 俞东进. 基于双层机器学习的业务流程剩余时间预测. *计算机学报*, 2021, 44(11): 2283–2294. [doi: [10.11897/SP.J.1016.2021.02283](https://doi.org/10.11897/SP.J.1016.2021.02283)]
- [25] 许行, 张凯, 王文剑. 一种小样本数据的特征选择方法. *计算机研究与发展*, 2018, 55(10): 2321–2330. [doi: [10.7544/issn1000-1239.2018.20170748](https://doi.org/10.7544/issn1000-1239.2018.20170748)]
- [26] 黄南天, 彭华, 蔡国伟, 徐殿国. 电能质量复合扰动特征选择与最优决策树构建. *中国电机工程学报*, 2017, 37(3): 776–785. [doi: [10.13334/j.0258-8013.pcsee.160108](https://doi.org/10.13334/j.0258-8013.pcsee.160108)]
- [27] 李占山, 姚鑫, 刘兆庚, 张家晨. 基于LightGBM的特征选择算法. *东北大学学报(自然科学版)*, 2021, 42(12): 1688–1695. [doi: [10.12068/j.issn.1005-3026.2021.12.003](https://doi.org/10.12068/j.issn.1005-3026.2021.12.003)]



郭娜(1996—), 女, 博士生, 主要研究领域为业务流程预测性监控, 流程挖掘, 机器学习.



陆婷(1990—), 女, 博士生, 主要研究领域为软件工程, 大规模的流媒体数据分析, 业务流程管理, Petri 网.



刘聪(1990—), 男, 博士, 教授, 博士生导师, CCF 专业会员, 主要研究领域为业务流程管理, 流程挖掘, 人工智能.



闻立杰(1977—), 男, 博士, 副教授, 博士生导师, CCF 专业会员, 主要研究领域为自然语言处理, 大数据处理与分析, 业务过程智能.



李彩虹(1970—), 女, 博士, 教授, 博士生导师, 主要研究领域为智能机器人, 计算智能学习算法.



曾庆田(1976—), 男, 博士, 教授, 博士生导师, CCF 高级会员, 主要研究领域为流程挖掘, Petri 网.

www.jos.org.cn

www.jos.org.cn