

一种可解释的云平台任务终止状态预测方法

刘春红^{1,2} 李为丽¹ 焦洁¹ 王敬雄¹ 张俊娜^{1,2}

¹(河南师范大学计算机与信息工程学院 河南新乡 453007)

²(智慧商务与物联网技术河南省工程实验室(河南师范大学) 河南新乡 453007)

(lch@htu.edu.cn)

An Interpretable Cloud Platform Task Termination State Prediction Method

Liu Chunhong^{1,2}, Li Weili¹, Jiao Jie¹, Wang Jingxiong¹, and Zhang Junna^{1,2}

¹(College of Computer Information Engineering, Henan Normal University, Xinxiang, Henan 453007)

²(Henan Provincial Engineering Lab of Intelligence Business & Internet of Things (Henan Normal University), Xinxiang, Henan 453007)

Abstract Based on feature selection and model interpretable method, a cloud platform task termination state prediction model with strong interpretability is constructed. The model visualizes the mapping relationship between static and dynamic attributes of tasks/jobs and termination status, then finds out the mapping mechanism between the load characteristics and the task termination states. The workload monitoring log published by Google is used, and the task dynamic information in the cloud platform is added. Shapley Additive explain (SHAP) is used to find out the importance of the influence of static and dynamic attributes on the termination state, and the results of task termination states prediction model modeling are explained by using the importance of variables combined with SHAP value and XGBoost model. Visualization technology is used to show how load characteristics affect the model's prediction of different task termination states. The average value of the absolute value of SHAP is used to measure the importance of features, and the global visualization of the importance of features in different termination states is realized. According to the results, 20 variables that have great influence on the prediction model of task termination states are selected as the basis of feature selection. How the change of characteristics affects the different termination states of tasks is visualized. From the visualization results, it can be seen that in the process of task running, different eigenvalues of each feature have influences on the termination states of the task, and different eigenvalues have different influences on the termination states. Feature selection combined with model interpretable method is applied to the construction process of task termination states prediction model, which can help to build a task termination states prediction model with high classification performance and easy understanding. By exploring the mapping mechanism between load characteristics and task termination status, the scheduling mechanism of cloud platform can be optimized.

Key words feature selection; termination statuses; global visualization; interpretability; mapping mechanism

摘要 基于特征选择和模型可解释方法构建可解释性强的云平台任务终止状态预测模型,该模型可视化任务/作业的静态和动态属性与终止状态之间的映射关系,进而找出负载特征与任务终止状态之间的映射机理。利用Google公开的工作负载监控日志,并加入云平台中任务的动态信息,采用沙普利加和解释(Shapley additive explain, SHAP)找出静态和动态属性对终止状态影响的重要性,利用变量重要性结合

收稿日期: 2022-09-06; 修回日期: 2023-06-26

基金项目: 国家自然科学基金项目(61902112); 河南省科技攻关项目(222102210047); 广西密码与信息安全重点实验室开放项目(GCIS202115)

This work was supported by the National Natural Science Foundation of China (61902112), the Key Science and Technology Research Project of Henan Province (222102210047), and the Open Project of Guangxi Key Laboratory of Cryptography and Information Security (GCIS202115).

SHAP 值和 XGBoost 模型, 对任务终止状态预测模型建模后的结果进行解释, 使用可视化技术呈现负载特征如何影响模型对不同任务终止状态的预测。用 SHAP 值绝对值的平均值衡量特征的重要性, 实现任务不同终止状态特征重要性的全局可视化, 根据结果筛选出对任务终止状态预测模型影响大的 20 个变量, 作为特征筛选的依据; 由可视化的结果可知, 任务运行过程中, 各特征的不同特征值对任务的终止状态有影响, 不同特征值对终止状态的产生有不同的影响。特征选择结合模型可解释性方法运用于任务终止状态预测模型的构建流程中, 可辅助构建高分类性能及易于理解的任务终止状态预测模型, 通过对负载特征与任务终止状态之间映射机理的探索, 可以优化云平台的调度机制。

关键词 特征选择; 终止状态; 全局可视化; 可解释性; 映射机理

中图法分类号 TP391

近年来, 机器学习模型在众多领域中都得到了广泛的应用, 并且表现出很好的性能。然而模型的黑盒特性, 使其缺乏透明度、可解释性和可信赖性, 如果不能从人类的角度对模型决策进行理解, 寻找更好的模型最终将沦为试错法。例如在云平台中, 对任务的终止状态进行预测时, 需要很高的透明度和可信度, 这意味着我们需要能够解释模型的决定、预测和证明它们的可靠性。这需要更好的可解释性, 意味着我们需要理解算法背后的机制^[1]。

在分布式云平台的调度管理中, 用户的应用请求以作业的形式提交到数据中心, 每个作业由调度器分配计算和存储资源, 作业的一次调度运行称为任务。作业包含 1 个或多个任务。其中, 任务终止状态包含驱逐、失败、完成和被杀^[2]。Jassas 等人^[3]提出了一种能够在早期发现失败作业的失败预测框架, 该框架的优势是减少资源浪费, 提高云应用的性能。Gao 等人^[4]提出了一种基于多层次双向长短期记忆(Bi-LSTM)的失败预测算法来识别云中的任务和作业失败, 提高以往基于机器学习和深度学习方法的失败预测精度。现有任务终止状态预测模型在预测精度上有很大的提高, 但是模型缺乏可解释性, 人们无法确定模型产生的输出是否符合常理, 所以即使模型产生较好的结果, 人们依然无法完全信任模型。

为了克服模型缺乏可解释性的限制, 必须考虑

用于建模的输入变量的可解释性, 即特征选择的可解释性, 以及模型预测的可解释性。由于模型内部复杂的结构, 会导致输入特征和预测结果之间的因果关系难以理解, 所以模型是不可解释的。基于此, 可以通过对特征进行统计分析, 以建立特征和输出之间的因果关系, 从而实现模型内部的透明化^[5-10]。特征统计分析方法是指对模型的特征进行汇总分析或者显著性可视化, 对混乱的特征进行统计分析, 计算不同特征对模型输出的贡献, 并对显著特征进行可视化。该方法是基于特征对模型进行解释, 特征作为可解释性和模型之间的桥梁^[11]。鉴于特征选择对模型可解释性的作用, 故研究基于树模型探讨其在任务终止状态预测研究中的实用性。

基于树的机器学习模型是当今最流行的非线性模型。随机森林、梯度增强树、决策树等树模型被用于金融、医学、生物学、客户保留、广告、供应链管理、制造、公共卫生和许多其他领域。基于输入特征集做出预测, 为每个输入特征分配信用数字来度量局部解释。如图 1 所示, 在任务终止状态预测中, 将一组相同的特征变量分别输入到黑盒模型和白盒模型, “白盒”局部解释可以通过为每个特征分配一个特征重要性系数来表示特征对于模型产生某种预测结果的影响程度。在保证模型预测性能的同时, 提高模型的可解释性, 其中的可解释性意味着我们可以

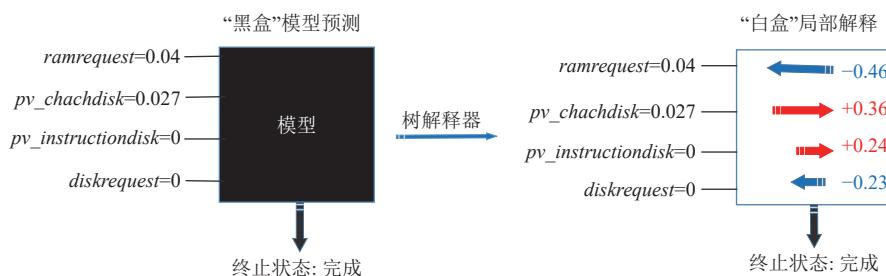


Fig. 1 Illustration of local explanation

图 1 局部解释示意图

理解模型如何使用输入特征来表示预测^[12].

然而, 尽管树的全局解释方法有丰富的历史, 总结了输入特征对整个模型的影响, 但是人们对解释输入特征对个体预测影响的局部解释的关注较少.

目前有 3 种比较常用的方法用来解释树模型的个体预测, 即局部解释方法: 1) 报告决策路径; 2) 一个未发表的启发式方法, 为每个输入特性分配积分; 3) 各种与模型无关的方法, 需要为每种解释多次执行模型^[12]. 这 3 种方法有 3 个局限性: 1) 简单地报告预测的决策路径对大多数模型都是没有帮助的, 特别是对那些基于多棵树的模型; 2) 行为启发式信用分配方法还没有被仔细分析, 对基于树的深度改变特征的影响有强烈的偏差; 3) 由于模型不可知的方法依赖于任意函数的事后建模, 它们可能速度很慢, 并存在抽样可变性.

树解释器, 是一种新的用于树的局部解释方法, 它使得最优局部解释的易处理计算成为可能, 由博奔论的理想性质定义^[13]. 研究借助任务终止状态预测模型, 选择出最佳特征子集, 并对进一步的模型可解释性进行研究. 为了解决当前树集成方法存在的不一致问题, 即当一个特征的真正影响实际增加时, 它们会降低该特征的重要性, 我们使用博奔论的最新应用, 将树集成特征属性方法与 SHAP 方法相结合, 使用 SHAP 值来衡量特征的重要性^[14], 将各特征对输出的影响可视化. 通过结合许多局部解释, 可以表示任务终止预测全局结构, 同时保持对原始模型的局部忠实性, 任务终止状态预测可解释性全局结构图如图 2 所示.

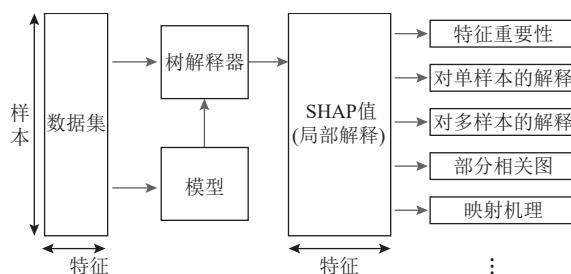


Fig. 2 Architecture of the task termination state prediction interpretability

图 2 任务终止状态预测可解释性架构

云平台中, 对任务的终止状态进行预测, 对预测结果采取相应的调度措施, 将失败的作业/任务提前终止, 以确保良好的时间性能, 并大量降低系统的资源浪费, 提高资源利用率. 通过解释作业/任务的系统属性、动态负载变化情况等因素与任务终止状态之间的关系, 可视化负载特征对预测结果的贡献度, 选

择重要特征, 进而提高任务终止状态预测模型的可解释性, 可以更好地调度任务.

本文的贡献包括 2 个方面:

1) 提出了一种可解释性强的云平台任务终止状态预测模型, 研究负载特征与任务不同终止状态之间的关系.

2) 通过一系列直观的可视化结果, 分析负载特征如何影响模型对任务不同终止状态的预测, 探索特征与任务终止状态之间的映射机理.

本文使用谷歌数据集进行实验, 在保证任务终止状态预测模型性能的同时, 打破模型的黑盒特性, 提高可解释性. 在下文中, 我们将讨论为什么树形模型在许多情况下都是最合适的模型, 因为它们的准确性及其可解释性(1.2 节), 将计算精确 SHAP 值的复杂度从 $O(T \times L \times 2^M)$ 降至 $O(T \times L \times D^2)$ (1.3 节), 分析全局特征重要性(2.2 节), 讨论对基于树的模型进行更好的局部解释的必要性(2.3 节和 2.4 节), 扩展了局部解释来捕获交互效应(2.5 节), 并找出特征与任务终止状态之间的映射机理(2.6 节).

1 可解释的任务终止状态预测方法

1.1 特征获取

Google 日志中包含任务各种资源的实际使用情况, 其能基本准确反映出任务在实际运行过程中消耗的资源, 但在早期运行过程中, 任务的数据所含信息量很少, 所以需要对数据进行额外的信息补偿, 以弥补模型预测中动态信息的不足, 动态信息主要与任务的动态属性有关. 在早期阶段, 任务的运行状态数据较少, 例如样本 CPU 使用率、分配的内存使用率、未映射的页面高速缓存内存使用率、平均磁盘 I/O 时间以及每个周期指令 (CPI) 等动态属性. 只利用单个任务数据构建预测模型, 限制了预测性能的提高. 同时, 部分作业之间具有相似资源消耗周期性变化的特点, 引入更多的作业之间的相似信息以增加预测模型的动态信息^[15].

加入任务的动态信息, 最终选择用于任务终止状态预测的 59 个特征^[16], 其包含有 5 个静态特征和 54 个动态特征. 特征描述如表 1 所示, 其中静态特征指不随着时间戳的变化而改变的特征, 动态特征指随着时间戳的变化而改变的特征.

为了描述方便, 本文使用英文特征名简写表示表 1 中的中文特征名称: 任务采样 CPU 使用率最大值 max_cpu , 任务规范内存使用的最大值 max_dsik , 任

Table 1 Static and Dynamic Characteristics of Tasks

表 1 任务的静态和动态特征

属性	特征描述
静态特征	作业调度类、任务优先级、任务的 CPU 资源请求、任务的 RAM 资源请求、任务的局部磁盘空间资源请求
	任务采样 CPU 使用率的最大值、均值、峰值、标准差、均方根、方根幅值
	任务规范内存使用的最大值、均值、峰值、标准差、均方根、方根幅值
	任务分配内存使用率的最大值、均值、峰值、标准差、均方根、方根幅值
	任务分映射页面缓存的内存使用的最大值、均值、峰值、标准差、均方根、方根幅值
	任务平均 disk 的 I/O 时间的最大值、均值、峰值、标准差、均方根、方根幅值
动态特征	任务平均使用的本地磁盘空间的最大值、均值、峰值、标准差、均方根、方根幅值
	任务每个指令存储器访问的最大值、均值、峰值、标准差、均方根、方根幅值
	任务每个指令周期的最大值、均值、峰值、标准差、均方根、方根幅值
	任务平均 CPU 的使用率的最大值、均值、峰值、标准差、均方根、方根幅值

务分配内存使用的最大值 $\max_fendsik$, 任务分映射页面缓存的内存使用的最大值 $\max_chachsik$, 任务平均 disk 的 I/O 时间的最大值 \max_dsikOI , 任务平均使用的本地磁盘空间的最大值 $\max_localdsik$, 任务每个指令存储器访问的最大值 $\max_instructiondsik$, 任务每个指令周期的最大值 $\max_perInstructiondsik$, 任务平均 CPU 的使用率的最大值 \max_Mcpu . 对于均值 ($mean$)、峰值(pv)、标准差(sd)、均方根(msr)、方根幅值(rpv)的命名同理.

1.2 基于 SHAP 的特征分析

树解释器可以通过平均许多局部解释作为一种全局方法^[13]. 如果对数据集中的所有样本都采用树解释器, 那么就可以得到一个特征重要性的全局度量, 它不会受到经典增益方法不一致性的影响, 与排列方法不同, 它不会错过高阶交互效应. 基于树解释器的全局特征归因, 与目前最先进的方法相比, 在存在交互作用时具有更高的检测重要特征的能力, 这对于基于树集合的特征选择任务具有重要意义.

为了更好地理解模型的输出, 可以从特征的角度出发对模型进行解释, 采用 SHAP 可视化特征对终止状态的重要性, 利用变量重要性结合 SHAP 值来解释 XGBoost 模型, 对任务终止状态预测模型建模后的结果进行解释. SHAP 值用于特征的重要性, 被定义为观察每个输入特征的值对模型输出顺序的影响, 并在所有可能的特征排序上取平均值^[2]. 对于所有可能的排序, 我们在模型输出的条件期望中每次引入一个特征, 然后将期望中的变化归因于所引入的特性.

SHAP 交互值由一个特征属性矩阵(对对角线的主要影响和对非对角线的交互影响)组成, 并具有类似于 SHAP 值的唯一性保证. 通过单独考虑单个模型预测的主要影响和交互效应, 树解释可以发现可能被遗漏的重要模式.

集成树模型做分类任务时, 模型输出的是一个

概率值. SHAP 实际是将输出值归因到每一个特征的 SHAP 值上, 依此来衡量特征对最终输出值的影响. SHAP 属于加性特征归因方法, 其表示如式(1)所示:

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z'_i. \quad (1)$$

其中 g 是解释模型; M 是输入特征的数量; $z'_i \in \{0, 1\}^M$ 表示在所有的 M 个特征中, 有多少特征是该样本所在的决策路径中包含的特征, 对于某个样本, 如果特征 k 不在其决策路径中, 那么对应特征的 SHAP 值为 0, 即 $\phi_k = 0$, 表示该特征不会对样本产生归因, 对于最终预测值没有贡献.

在树模型中, 对于某个特征 j , 需要针对所有可能的特征组合(包括不同顺序)计算 SHAP 值, 然后进行加权求和, 如式(2)所示:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(M-|S|-1)!}{M!} [f_x(S \cup \{i\}) - f_x(S)]. \quad (2)$$

其中 N 为训练集中所有特征的集合; M 为特征数量; S 是模型中使用的特征的子集; $f_x(S)$ 表达的是只利用特征集合 S , 根据树的结构、叶子节点的取值、内部节点的 Cover 值等, 计算出样本的平均值; $f_x(S \cup \{i\})$ 表达的是在特征集合的基础上, 加上特征 i , 然后再根据树的结构、叶子节点的取值、内部节点的 Cover 值等, 计算出样本的平均值; $\frac{|S|!(M-|S|-1)!}{M!}$ 是对应特征子集 S 下, 对于上述包含特征 i 和不包含特征 i 的情况下, 样本取值之差的权重.

正如 Lundberg 等人^[12] 所提到的, 这类附加特征属性方法的一个重要属性是, 该类方法中存在一个唯一的解, 且该解具有 3 个期望的属性: 局部精确性、一致性和缺失性. 局部精确性表示特征归因的总和等于我们要解释的模型的输出; 一致性表明, 改变模型以使某个特性对模型产生更大的影响, 永远不会减少分配给该特征的属性的总和等于我们试图解释

的函数的输出;缺失性表示已经缺失的特征(例如 $z'_i = 0$)不具有重要性.

如果忽略计算复杂度,那么可以通过估计 $E[f(x)|x_S]$ 计算树的SHAP值,然后使用式(2),其中 $f_x(S) = E[f(x)|x_S], E[f(x)|x_S]$ 是输入特征的子集 S 的条件期望值,

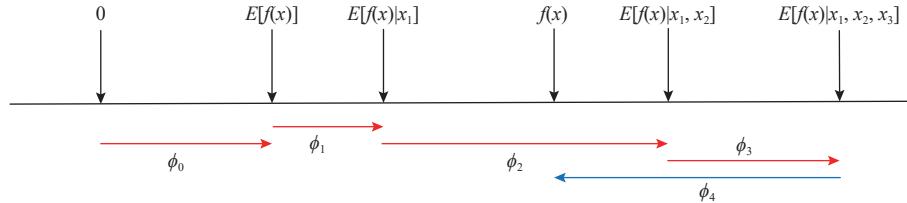


Fig. 3 Principle of SHAP addition explanation

图3 SHAP 加和解释原理

1.3 融合 SHAP 的可解释任务终止状态预测算法

树解释器用于任务终止状态可解释性研究中,其背后的主要算法可以精确地计算SHAP值,从而保证一致的解释.在低阶多项式时间内精确计算式(2),其中条件期望函数 f_x 是使用树遍历(算法1)来定义的.通过直接计算SHAP值,我们能够保证解释总是-致的和局部精确的.

在 $O(T \times L \times 2^M)$ 时间内直接估计SHAP值,对于一个树形模型, $E[f(x)|x_S]$ 可以使用算法1进行递归估计.算法1中 v 是节点值的向量,它取内部节点的值;向量 a 和 b 分别是每个内部节点的左右节点索引;向量 t 是每个内部节点的阈值;而 d 是用于在内部节点中分割的特征的索引向量;向量 r 表示多个数据样本落在该节点中,样本数量为向量的分量个数;权重 w 表示与子集 S 匹配的训练样本落在节点的比例.算法1中,创建程序 G 来实现整个递归过程,算法的复杂度为 $O(T \times L \times 2^M)$.

算法1. 估计 $E[f(x)|x_S]$.

```
procedure EXPVALUE(x,S,tree={v,a,b,t,r,d})
  procedure G(j,w)
    if  $v_j \neq \text{internal}$  then
      return  $wv_j$ ;
    else
      if  $d_j \in S$  then
        return  $G(a_j, w)$  if else  $G(b_j, w)$ ;
      else
        return  $G(a_j, wr_{a_j}/r_j) + G(b_j, wr_{b_j}/r_j)$ ;
      end if
    end if
  end procedure
end procedure
```

件期望值,图3解释了如何从 $E[f(x)]$ 得到预测值,SHAP值将每个特征的归因值赋值为在调整该特征时模型预测的预期变化,将模型 f 对于样本 $x_1 = a_1, x_2 = a_2, x_3 = a_3, x_4 = a_4$ 的预测解释为引入条件期望的每个特征的影响 ϕ_j 的总和.

return $G(1, 1)$;

end procedure

在 $O(T \times L \times D^2)$ 时间内直接估计SHAP值,将对树与树之和的精确SHAP值的计算复杂度从指数多项式降低到低阶多项式.具体来说,时间复杂度降为 $O(T \times L \times D^2)$,空间复杂度为 $O(D^2 + M)$,同时在树为平衡树的时候,深度 D 变成 $\log L$.其中 L 是树中的最大叶子数量, M 是特征数量. m 是我们到目前为止已经分割过的唯一特征的路径,它包含4个属性:1) d 是特征索引;2) z 是流经这个分支的“0”路径,即这个特征不在特征子集 S 中的概率;3) o 是通过这个分支的“1”路径,即这个特征在特征子集 S 中的概率;4) w 是用来保持给定基数集合的比例.我们使用点表示法来访问这些成员,对于整个向量, $m.d$ 表示所有特征索引的向量.算法描述如算法2所示,每次调用RECURSE都会导致内部节点的复杂度为 $O(D)$,或者叶节点的复杂度为 $O(D^2)$,因为展开嵌套在以 D 为界的循环中.这导致整个树的复杂度为 $O(L \times D^2)$,因为在叶节点完成的工作决定了内部节点的复杂度,对于整个树集合 T ,这个界限变成 $O(T \times L \times D^2)$.

算法2. Tree SHAP.

```
procedure TS(x, tree = {v,a,b,t,r,d})
   $\phi = \text{array of } len(x) \text{ zeros};$ 
  procedure RECURSE(j,m,p_z,p_o,p_i)
     $m = EXTEND(m, p_z, p_o, p_i);$ 
    if  $v_j \neq \text{internal}$  then
      for  $i \leftarrow 2 \text{ } len(m)$  do
         $w = sum(UNWIND}(m,i).w);$ 
         $\phi_{m_i} = \phi_{m_i} + w(m_i.o - m_i.z)v_j;$ 
      end for
    end if
  end procedure
end procedure
```

```

else
     $h, c = x_{d_j} \leq t_j ? (a_j, b_j) : (b_j, a_j);$ 
     $i_z = i_o = 1;$ 
     $k = \text{FINDFIRST}(\mathbf{m}, d_j);$ 
    if  $k \neq \text{nothing}$  then
         $i_z, i_o = (m_{k,z}, m_{k,o});$ 
         $\mathbf{m} = \text{UNWIND}(\mathbf{m}, k);$ 
    end if
     $\text{RECURSE}(h, \mathbf{m}, i_z, r_h / r_j, i_o, d_j);$ 
     $\text{RECURSE}(c, \mathbf{m}, i_z, r_c / r_j, 0, d_j);$ 
end if
end procedure

procedure  $\text{EXTEND}(\mathbf{m}, p_z, p_o, p_i)$ 
     $l = \text{len}(\mathbf{m});$ 
     $\mathbf{m} = \text{copy}(\mathbf{m});$ 
     $m_{l+1}.(\mathbf{d}, z, o, w) = (p_i, p_z, p_o, l = 0?1:0);$ 
    for  $i \leftarrow l-1$  do
         $m_{i+1}.w = m_{i+1}.w + p_o m_i.w(i/l);$ 
         $m_i.w = p_z m_i.w[(l-i)/l];$ 
    end for
    return  $\mathbf{m};$ 
end procedure

procedure  $\text{UNWIND}(\mathbf{m}, i)$ 
     $l = \text{len}(\mathbf{m});$ 
     $n = m_l.w;$ 
     $\mathbf{m} = \text{copy}(m_1 \dots l-1);$ 
    for  $i \leftarrow l-1$  do
        if  $m_i.o \neq 0$  then
             $t = m_j.w;$ 
             $m_j.w = n \cdot l / (j \cdot m_i.o);$ 
             $n = t - m_j.w \cdot m_i.z((l-j)/l);$ 
        else
             $m_j.w = (m_j.w \cdot l) / (m_i.z(l-j));$ 
        end if
    end for
    for  $j \leftarrow i$  to  $l-1$  do
         $m_j.(\mathbf{d}, z, o) = m_{j+1}.(\mathbf{d}, z, o);$ 
    end for
    return  $\mathbf{m};$ 
end procedure

 $\text{RECURSE}(1, [], 1, 1, 0);$ 
return  $\emptyset;$ 

```

end procedure

传统的特征重要性方法可以直观地反映出特征的重要性,但是依然无法得出特征与模型预测结果之间的关系,SHAP属于加性特征归因方法,将每个特征都视为贡献者,计算每个特征的贡献值,可以反映出每个特征对于最终预测结果的影响,增加任务终止状态预测模型的可解释性。使用XGBoost模型对任务终止状态进行预测,从特征的角度出发,以SHAP值来衡量模型预测中特征的重要性,并对其结果进行直观的可视化。使用SHAP总结图可视化特征的变化如何影响任务的不同终止状态,进而找出特征与任务终止状态之间的映射机理。

2 实验及结果分析

2.1 数据集和评价指标

本文以Google云平台的计算调度系统Borg^[17]为研究对象,利用Google公开的工作负载监控日志数据集^①进行验证,该日志有1个月的监控数据。验证实验筛选了Google集群日志前3天的数据进行实验。该样本数据集中共有10 473个job,包含1 665 280个任务。为了更准确地对云平台任务不同终止状态的影响因素进行更直观的可视化,需要进一步对任务的终止状态进行细分,即不仅预测任务是成功(finish)还是失败,还需要进一步细分失败是驱逐(evict)、失败(fail)或被杀(kill)的状态,从而采取提前终止或继续运行等不同的调度策略。

由于Google云平台数据集是从真实环境中采集获取,同时数据集存在含有部分噪声和缺失值等问题,所以需要对数据进行清洗预处理。实验筛选了Google集群日志前3天的数据,处理清洗过程参考文献[17]。

多分类评价指标分为宏平均(macro-average)和微平均(micro-average),以召回率(recall)为例,Macro-R表示宏平均的召回率,Micro-R表示微平均的召回率。针对极度不均衡的多分类来说,Macro-R受样本数量少的类别影响较大,比Micro-R更合理^[15]。所以本文在测试数据集上,选择宏平均来度量分类器性能,更侧重对小类别的有效性,本文使用的评价指标均使用宏平均的方法进行计算。

Kappa评价指标 k 可以用来进行多分类模型准确度的评估, k 的取值范围是[-1,1],实际应用中一般

^① <https://github.com/google/cluster-data>

$k \in [0,1]$, 与 ROC 曲线中一般不会出现下凸形曲线的原理类似. k 的值越高, 则代表模型实现的分类准确度越高.

海明距离用来衡量预测标签与真实标签之间的距离, 取值在 0~1 之间. 海明距离为 0 说明预测结果与真实结果完全相同, 海明距离为 1 说明模型与我们想要的结果完全背道而驰.

各个多分类各评价指标的计算如式(3)~(6)所示. 其中 $Macro-R$ 表示召回率的宏平均; 召回率表示任务样本中有多少运行成功的样本被正确预测; $Macro-P$ 表示准确率的宏平均; $Macro-F1$ 表示 $F1$ 的宏平均; k 表示 Kappa 系数, 式(6)中 p_o 是总体精度, p_e 是偶然一致性误差.

$$Macro-R = \frac{1}{n} \sum_{i=1}^n R_i, \quad (3)$$

$$Macro-P = \frac{1}{n} \sum_{i=1}^n P_i, \quad (4)$$

$$Macro-F1 = 2 \times \frac{Macro-R \times Macro-P}{Macro-R + Macro-P}, \quad (5)$$

$$k = \frac{p_o - p_e}{1 - p_e}. \quad (6)$$

实验设计为 2 组: 1) 特征重要性的可视化, 使用特征重要性图表示 4 种不同的任务终止状态的特征重要性; 2) 特征对模型输出影响的可视化, 使用个体样本力图、多个样本力图、部分相关图、SHAP 总结图可视化特征如何影响任务的不同终止状态.

2.2 特征重要性分析

绘制树集合模型中特征的影响通常是用条形图来表示全局特征重要性, 或用部分相关性图来表示单个特征的影响. 然而, 由于 SHAP 值是个性化的特征属性, 对于每个预测都是唯一的, 因此它能够实现新的、更丰富的视觉表示.

取每个特征的 SHAP 值绝对值的平均值作为该特征的重要性, 并进行降序排序, 由于任务有多种终止状态, 得到的结果是堆叠的条形图, 基于 XGBoost 模型的 20 个重要变量的排序结果如图 4 所示.

从图 4 可以看到, 在选出的 20 个重要特征中, finish 类不使用 $sd_instructiondisk$, pv_cpu , sd_cpu , $max_$

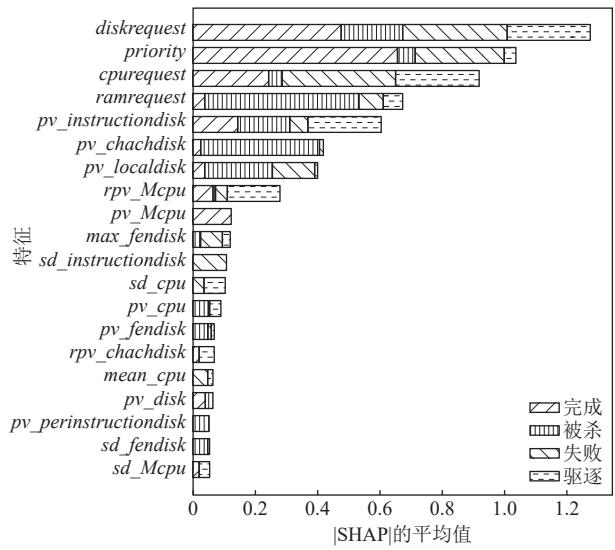


Fig. 4 Feature importance map

图 4 特征重要性图

$fendisk$ 等特征; kill 类不使用 pv_Mcpu , sd_cpu , sd_Mcpu , $rpv_chachdisk$, $mean_cpu$ 这些特征; fail 类不使用 pv_Mcpu , $rpv_chachdisk$, $rpv_chachdisk$, pv_disk 这些特征; evict 类不使用 pv_disk , pv_Mcpu , $sd_instructiondisk$, $pv_chachdisk$ 等特征.

2.3 对单个样本的解释

对于正确分类的样本, 通过单个样本的力图, 可以得知模型预测值的高低是如何造成的、由哪些特征造成. 如图 5 所示, 模型 $predict_proba$ 值为 0.91, 代表当前样本的 SHAP 值的输出值, 基值为 0.4298, 即全体样本的平均值, 模型作出这样的结果, 主要是受到任务平均 CPU 使用率的方根幅值、任务采样 CPU 使用率的方根幅值、任务平均 CPU 使用率的均方差、任务采样 CPU 使用率的最大值的影响, 这些特征对模型产生正向作用. 任务分映射页面缓存的内存使用的最大值、任务分映射页面缓存的内存使用的峰值是对模型产生反向作用的特征.

根据模型的预测结果找出被模型分错的样本, 对其进行可解释性分析, 查看哪些因素导致样本被错误分类. 在测试集中, 样本 ID 为 2610 的样本是成功完成的, 但是模型将其终止状态预测为失败, 导致模型这种错误预测的原因如图 6 所示, 主要受到任

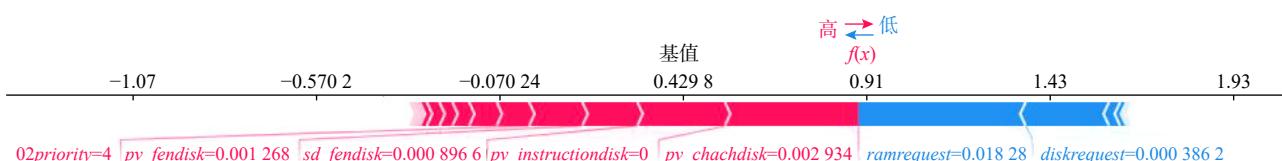


Fig. 5 Force plot of prediction correct sample

图 5 预测正确的样本力图

务平均 CPU 的使用率峰值、任务磁盘空间资源请求、任务每个指令存储器的访问峰值、任务分映射页面缓存的内存使用峰值的影响.

2.4 对多个样本的解释

对多个样本的预测结果进行可视化，可以将单个样本的力图旋转 90°，然后横向堆叠，得到力图的变体，这样我们可以得到对整个数据集的解释。对于多个样本的解释，在生成的力图中，可以选择不同的

横纵坐标，实现对模型预测的直观可视化。从图 7 可以看出，在基于样本的解释中，对输出贡献正向影响的特征和对输出贡献负向影响的特征； x 轴是样本数量， y 轴是 SHAP 值加总（每个特征的 SHAP 值）；左边红色扎堆是正向 SHAP 增益区，任务的局部磁盘空间资源请求、优先级对一些样本是正向增益的。整体来说，图 7 是一个宏观的了解，将诸多样本中不同特征对预测结果产生的影响进行可视化。

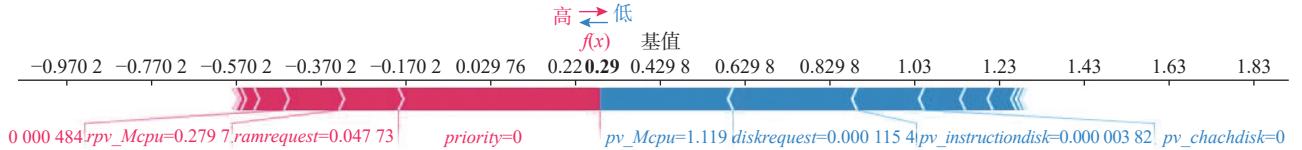


Fig. 6 Force plot of prediction wrong sample

图 6 预测错误的样本力图

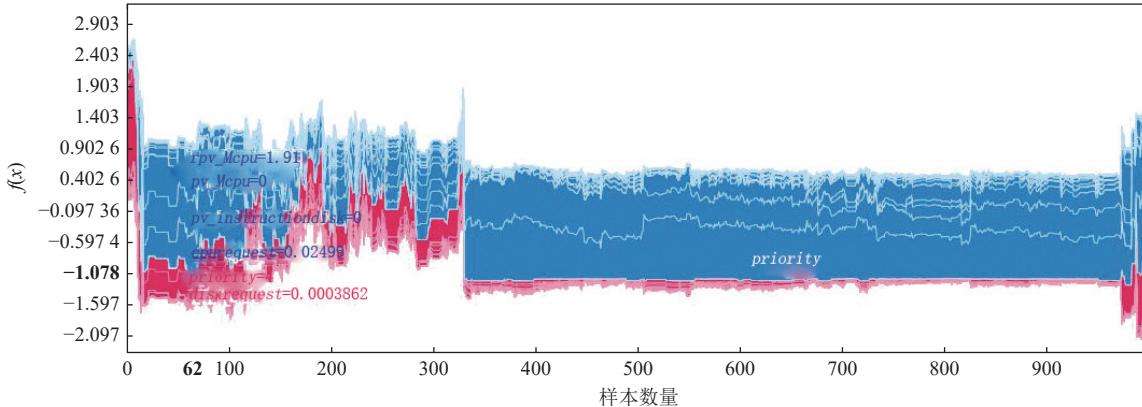


Fig. 7 Force plot of prediction multiple samples

图 7 预测多个样本的力图

2.5 特征相关图

特征相关图表示特定变量或变量组的值固定时模型的预期输出。固定变量的值是变化的，并绘制出最终的预期模型输出。绘制一个函数的预期输出如何随着我们改变的一个特征而改变，有助于解释模型如何依赖于该特征。

特征相关图如图 8 所示，可以更清楚地显示出特征对于模型输出的影响。 x 轴表示特征的取值， y 轴表示该特征的 SHAP 值，也就是说特征的取值会给模型的输出带来变化量。通过绘制数据集中许多任务的这些值，我们可以看到特征的属性重要性如何随其值的变化而变化。标准的相关图仅产生直线，而特征相关图由于模型中的相互作用而捕捉垂直离差。这些效果可以通过用交互特征的值给每个点着色来可视化。

对于相关图中的同一个 x 值，也就是特征取值相同的样本，它们的 SHAP 值不同。其原因是，该特征和

其他特征有着交互效应，相关图可以自动地选择与该特征相关的另外一个特征，来表现特征之间的这种交互效应。图 8 右边是对比的特征，红色代表相关特征的高分部分，蓝色代表相关特征的低分部分。任

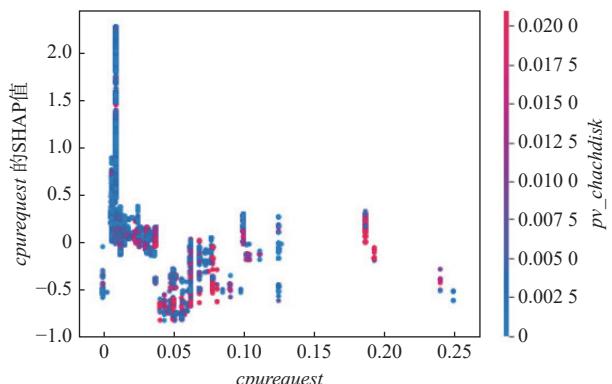


Fig. 8 Feature dependence plot

图 8 特征相关图

务分映射页面缓存的内存使用的峰值的高分部分,对于低任务的 CPU 资源请求来说, SHAP 值一般小于 0, 所以是负面影响.

2.6 特征与终止状态之间的映射机理

使用 SHAP 特征归因方法实现对特征与任务终止状态之间关系的解释, 对不同终止状态预测的特征重要性进行直观的可视化, 根据可视化的结果, 分别找出任务终止状态为完成、被杀、失败、驱逐的影响因素, 并对特征如何影响模型的预测结果进行分析, 找出特征与终止状态之间的映射机理.

特征重要性条形图给出了训练数据集中相对重要的概念, 但并不表示特征对模型输出的影响范围和分布, 也不表示特征的值与其影响的关系. SHAP 摘要图利用个性化的特征属性来传达特征重要性的所有方面, 同时保持视觉简洁.

SHAP 值类似回归系数, 有正负之分, 也有大小之分, SHAP 值越高, 表示在任务终止状态预测模型中预测为该类的对数的几率越高, 数据集中的每个样本都在模型中运行, 并且为每个特征属性值创建一个点, 因此每个任务在特征的线上获得一个点, 该点由该任务的特征值着色, 并垂直堆积以显示密度. 每个点都是一个特征和实例的 SHAP 值.

SHAP 概要图如图 9 所示, 其中 y 轴上的位置由特征决定, x 轴上的位置由对应特征的 SHAP 值决定. 图 9 的右侧有一条表示特征值大小的线, 颜色从蓝色变为红色, 代表特征的值从低到高. 如果特征对模型输出的影响随着其值的变化而平滑变化, 则该颜色也将具有平滑的渐变. 重叠点在 y 轴方向上抖动, 因此我们得到每个特征的 SHAP 值的分布, 这些特征是根据它们的重要性排序的.

由图 9(a) 可知, 对于终止状态为 finish 的任务, *priority* 这一特征对模型非常重要, 而且 *priority* 值较大时, SHAP 值小于 0, 对模型产生负向影响, 会降低输出为 finish 的概率. 横向来看, *diskrequest* 这个特征, 样本分布较为分散, 具有长长的右尾, 对该类的预测产生正向影响. 不难理解, 在云平台中, 过高优先级的任务需要被优先调度, 但在机器资源过度分配时, 其资源请求得不到满足, 容易失败.

由图 9(b) 可知, 对于终止状态为 kill 的任务, *ramrequest* 对模型输出的影响最大, 较大的 *ramrequest*, 使得 SHAP 值大于 0, 对模型产生正向影响, 增加模型产生这种输出的概率. 资源请求值表示在任务运行过程中, 各请求被允许使用的最大值, 当任务的资源请求超过资源请求值时, 该任务可能会被终

止或者受到限制. 当调度程序在计算机上过度提交资源时, 即使任务请求小于限制值, 也会出现没有足够的资源满足任务运行时请求的情况, 这种情况下, 云平台会杀死 1 个或多个低优先级任务.

由图 9(c) 可知, 对于终止状态为 fail 的任务, *cpurequest*, *diskrequest* 对模型输出的影响较大, 对于较小的 *cpurequest*, SHAP 值大于 0, 增加模型产生这种输出的概率. 在资源请求较少时, 任务在运行过程中有很大概率会超过资源请求的界限, 导致任务失败. 同时, *priority* 对类 fail 的影响也很大, 对于较高的 *priority*, SHAP 值大于 0, 增加模型产生这种输出的概率, 即过高优先级的任务更容易失败.

由图 9(d) 可知, 对于终止状态为 evict 的任务, *cpurequest*, *diskrequest*, *pv_instructiondisk*, *rpv_Mcpu*, *ramrequest* 对模型输出的影响较大. 当任务的实际需求超过计算机的处理能力时, 任务的资源请求得不到满足, 或磁盘所保存的任务数据丢失, 都可能会导致任务被驱逐.

通过 SHAP 摘要图, 可以直观清晰地看出任务不同终止状态的影响因素, 以及各个特征是如何影响任务的不同终止状态的, 通过对可视化的结果进行分析, 找出特征与终止状态时间的映射机理.

2.7 任务终止状态预测模型的选择

本节分别构建 XGBoost、随机森林、GBDT、朴素贝叶斯、决策树、 k 近邻算法、Adaboost 这 7 种任务终止预测模型. 由表 2 的模型性能评价结果可得, 除了朴素贝叶斯模型, 其余 6 个模型均具有较好的分类性能, 其中, XGBoost 模型具有较高的 Kappa 值 (0.686)、ACC 值 (0.9597)、Macro-P 值 (0.9564)、Macro-R 值 (0.9572)、Macro-F1 值 (0.9531), 并且其具有较低的海明距离值 (0.057). 综合考虑各个指标的最优值, 最终选择 XGBoost 模型作为任务终止状态预测模型, 该模型计算简单且易于理解, 并且具有较强的可解释性.

研究结果表明, 与已有工作相比, 在进行特征选择和处理的基础上, 本文所提的融合 XGBoost 与 SHAP 模型的可解释性方法可以更好地对云平台中任务的终止状态进行预测.

3 结束语

大型异构云计算平台上, 使用特征选择结合模型可解释性方法, 构建易于理解的任务终止状态预测模型. XGBoost 模型结合 SHAP 方法, 在保证模型

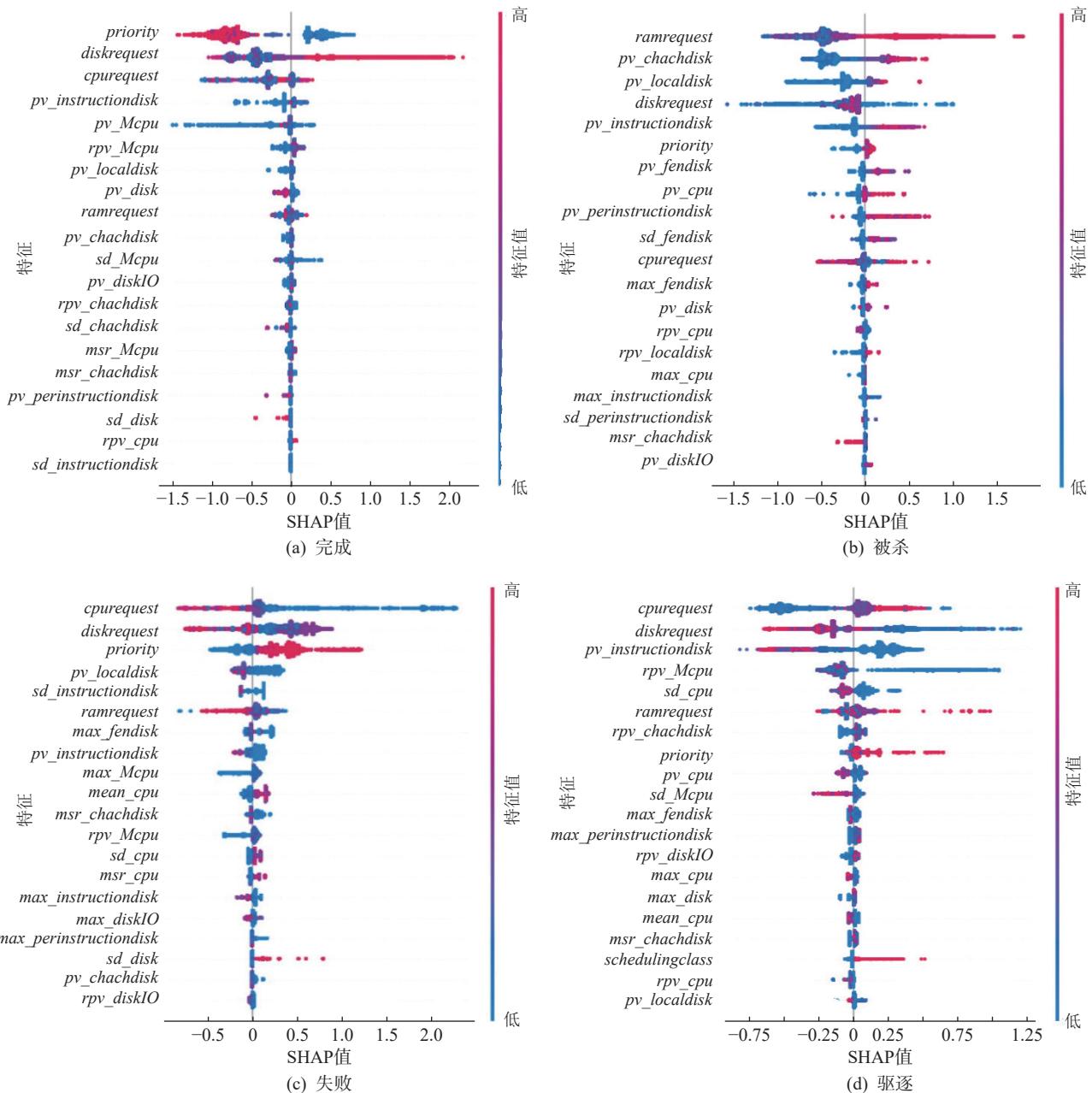


Fig. 9 SHAP summary chart of different task termination states

图 9 不同任务终止状态的 SHAP 概要图

Table 2 Comparison of Prediction Performance of Different Models

表 2 不同模型预测性能的对比

模型	运行时间/s	Kappa	海明距离	ACC	Macro-P	Macro-R	Macro-F1
XGBoost(本文)	2 673	0.686	0.057	0.9572	0.9564	0.9572	0.9531
随机森林	631.56	0.3095	0.2045	0.9326	0.9346	0.9326	0.9177
GBDT	16 677	0.5090	0.1147	0.9547	0.9552	0.9548	0.9496
朴素贝叶斯	7.76	0.0878	0.3895	0.6185	0.8566	0.6158	0.7174
决策树	50.63	0.2442	0.2468	0.9292	0.9339	0.9292	0.9191
k 近邻算法	4.9283	0.2442	0.2468	0.9396	0.9390	0.9396	0.9294
Adaboost	4 256.5	0.0844	0.3976	0.8311	0.9173	0.8311	0.8455

预测性能较好的同时,提高模型的可解释性。从特征的角度出发,对任务的不同终止状态进行可解释性研究,可视化特征与终止状态之间的关系,对于实验结果进行了深入分析,找出特征是如何影响不同任务终止状态的,进而探索出特征与任务终止状态之间的映射机理,实现对云平台系统调度决策的优化,提升云平台的运算性能。未来将继续研究云边融合中任务终止状态可解释性方面的工作,探索更好的可解释性方法,并对负载特征与任务终止状态之间的映射机理进行更加深入的探索,增加本文所提方法的应用范围。

作者贡献声明: 刘春红提出方法思路和实验方案;李为丽负责完成实验并撰写论文;焦洁提出特征选择意见;王敬雄协助完成实验;张俊娜提出论文整体的修改意见。

参 考 文 献

- [1] Tjoa E, Guan C. A survey on explainable artificial intelligence : Toward medical[J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 32(11): 4793–4813
- [2] Verma A, Pedrosa L, Korupolu M , et al. Large-scale cluster management at Google with Borg[C]//Proc of the 10th European Conf on Computer Systems. New York: ACM, 2015 [2022-12-26].<https://doi.org/10.1145/2741948.2741964>
- [3] Jassas M S , Mahmoud Q H . Failure characterization and prediction of scheduling jobs in Google cluster traces[C/OL]// Proc of the 10th IEEE GCC Conf & Exhibition (GCC). Piscataway, NJ: IEEE, 2019 [2022-12-26].<https://ieeexplore.ieee.org/document/9087621>
- [4] Gao Jiechao, Wang Haoyu, Shen Haiying. Task failure prediction in cloud data centers using deep learning[J]. IEEE Transactions on Services Computing, 2020, 15(3): 1411–1422
- [5] Shrikumar A, Greenside P, Kundaje A. Learning important features through propagating activation differences[C]// Proc of Int Conf on Machine Learning. New York: ACM, 2017: 3145–3153
- [6] Ventura F, Cerquitelli T, Giacalone F. Black-box model explained through an assessment of its interpretable features[C]// Proc of European Conf on Advances in Databases and Information Systems. Berlin: Springer, 2018: 138–149
- [7] Fong R C, Vedaldi A. Interpretable explanations of black boxes by meaningful perturbation[C]//Proc of IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2017: 3429–3437
- [8] Petsiuk V, Das A, Saenko K. Rise: Randomized input sampling for explanation of black-box models[J]. arXiv preprint, arXiv: 1806.07421, 2018
- [9] Ma Liantao, Zhang Chaohe, Jiao Xianfeng, et al. Dr. Deep: Interpretable evaluation of patient health status via clinical feature's context learning[J]. Journal of Computer Research and Development, 2021, 58(12): 2645–2659(in Chinese)
(马连韬, 张超贺, 焦贤锋, 等. Dr. Deep: 基于医疗特征上下文学习的患者健康状态可解释评估 [J]. *计算机研究与发展*, 2021, 58(12): 2645–2659)
- [10] Baumgartner C F, Koch L M, Tezcan K C, et al. Visual feature attribution using wasserstein GANs[C]//Proc of IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2018: 8309–8319
- [11] Hua Yingying, Zhang Daichi, Ge Shiming. Research progress in the interpretability of deep learning models[J]. Journal of Cyber Security, 2020, 5(3): 1–12 (in Chinese)
(化盈盈, 张岱墀, 葛仕明. 深度学习模型可解释性的研究进展 [J]. *信息安全学报*, 2020, 5(3): 1–12)
- [12] Lundberg S M, Lee S I. A unified approach to interpreting model predictions[C]//Proc of the 31st Int Conf on Neural Information Processing Systems. New York: ACM, 2017: 4768–4777
- [13] Lundberg S M, Erion G, Chen H, et al. Explainable AI for trees: From local explanations to global understanding [J]. arXiv preprint, arXiv: 1905.04610, 2019
- [14] Lundberg S M, Erion G G, Lee S I. Consistent individualized feature attribution for tree ensembles[J]. arXiv preprint, arXiv: 1802.03888, 2018
- [15] Dai Liping. Research on prediction method of task termination status of large-scale cloud platform[D]. Xinxiang: Henan Normal University 2020(in Chinese)
(代丽萍. 大规模云平台任务终止状态预测方法研究 [D]. 新乡: 河南师范大学, 2020)
- [16] Liu Chunhong, Han Jingjing, Shang Yanlei. Predicting job failure in cloud cluster: Based on SVM classification[J]. Journal of Beijing University of Posts and Telecommunications, 2016, 39(5): 104–109(in Chinese)
(刘春红, 韩晶晶, 商彦磊. 基于 SVM 分类的云集群失败作业主动预测方法 [J]. 北京邮电大学学报, 2016, 39(5): 104–109)
- [17] Chen Xin, Lu C D, Pattabiraman K. Failure analysis of jobs in compute clouds: A Google cluster case study[C]// Proc of the 25th IEEE Int Symp on Software Reliability Engineering. Piscataway, NJ: IEEE, 2014: 167–177



Liu Chunhong, born in 1969. PhD, associate professor, master supervisor. Her main research interests include service calculation, cloud computing, and machine learning.

刘春红, 1969年生.博士, 副教授, 硕士生导师. 主要研究方向为服务计算、云计算、机器学习.



Li Weili, born in 1995. Master. Member of CCF. Her main research interests include cloud computing and research of machine learning interpretability.

李为丽,1995 年生.硕士.CCF 会员.主要研究方向为云计算、机器学习可解释性研究.



Wang Jingxiong, born in 1996. Master. Member of CCF. His main research interests include cloud computing, machine learning, and time series anomaly detection.

王敬雄,1996 年生.硕士.CCF 会员.主要研究方向为云计算、机器学习、时间序列异常检测.



Jiao Jie, born in 1996. Master. Member of CCF. Her main research interests include cloud computing, machine learning, and load sequence forecasting.

焦 洁,1996 年生.硕士.CCF 会员.主要研究方向为云计算、机器学习、负载序列预测.



Zhang Junna, born in 1979. PhD, associate professor. Member of CCF. Her main research interests include edge computing and service computing.

张俊娜,1979 年生.博士, 副教授.CCF 会员.主要研究方向为边缘计算、服务计算.