

文章编号:1007-5321(2019)05-0062-07

DOI:10.13190/j.jbupt.2018-308

# 一种在线集群异常作业预测方法

谢丽霞, 汪子荧

(中国民航大学 计算机科学与技术学院, 天津 300300)

**摘要:** 设计了作业子任务动态特征计算方式;其次依据此动态特征提出一种改进门控递归单元(IGRU)神经网络;然后采用IGRU根据动态特征实时预测任务终止状态是否异常;最后根据作业与其子任务运行状态之间的状态相关性检索异常作业,完成对异常作业的预测. 实验结果表明,在线集群异常作业预测在预测灵敏度、误差率、精确度和预测时长方面与其他预测方法相比有明显提升;在保障集群平台安全方面具有一定的应用性.

**关键词:** 集群异常作业; 动态特征; 实时预测; 门控递归单元; 状态相关性

**中图分类号:** TN911.22

**文献标志码:** A

## An Online Cluster Anomaly Job Prediction Method

XIE Li-xia, WANG Zi-ying

(School of Computer Science and Technology, Civil Aviation University of China, Tianjin 300300, China)

**Abstract:** An online cluster anomaly job prediction method (OCAJP) is proposed. Firstly, a calculation of dynamic features of sub-tasks in the job was designed. Secondly, an improved gated recurrent unit (IGRU) neural network was designed according to the dynamic features. Then, the IGRU was used to predict whether the sub-task's final status was abnormal according to its dynamic features. Finally, the anomaly job was obtained based on the status relevance between the job and its sub-tasks, so as to complete prediction of abnormal jobs. The experimental results showed that OCAJP had a significant improvement in prediction sensitivity, error rate, accuracy, and prediction time compared with other prediction methods; this method had applicability in protecting the security of the cluster platform.

**Key words:** cluster anomaly job; dynamic features; real-time prediction; gated recurrent unit; status relevance

由于集群平台处理集群作业的复杂性,集群作业异常频发. 统计分析 Google 在 2012 年发布的 2011 年 5 月数据中心资源负载日记<sup>[1]</sup>,将作业子任务的最终执行状态划分为正常执行与不正常执行两大类. 并且在总计超过 67 万个集群作业,包含了 2 600 个子任务中只有 57.6% 的作业被成功执行,超过 40% 的作业因发生异常无法得到执行结果<sup>[2]</sup>. 由于一个作业通常被划分成多个不同的

子任务由不同的虚拟机处理<sup>[3]</sup>,因此某一子任务的执行状况会对其所属作业的执行状态造成相应影响<sup>[4]</sup>. 而为了提高执行及计算效率,云集群计算平台会对无法正常执行的作业子任务进行反复调度<sup>[5]</sup>,且此类重调度没有上限次数. 这种对异常的作业子任务进行无限制的重调度会显著增加调度负载,导致系统资源剧烈变化,且调度受限,影响其他作业的执行. 这一点可能被攻击者利用,对云

收稿日期: 2018-12-23

基金项目: 国家自然科学基金民航联合研究基金项目(U1833107); 国家科技重大专项项目(2012ZX03002002); 中央高校基本科研业务费专项资金项目(ZYGX2018028)

作者简介: 谢丽霞(1974—),女,教授, E-mail: lxxie@126.com.

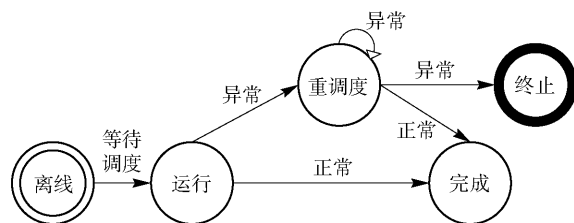
集群平台实施任务注入攻击或拒绝服务攻击<sup>[6]</sup>,为避免异常作业对集群系统及用户隐私产生危害,在作业执行前期准确、快速预测其终止状态的方法已成为研究热点。

目前作业终止状态在线预测方法的研究主要集中于分析作业特征与作业终止状态之间的联系。Sonoda等<sup>[7]</sup>采用朴素贝叶斯神经网络根据作业子任务执行时的消息日志在线判断此作业子任务是否属于异常任务,该方法未考虑作业状态与其子任务状态之间的相关性,因此预测时间较长。唐红艳等<sup>[8]</sup>提出一种异常频率阈值学习(AFTL, abnormal frequency threshold learning)算法计算作业执行前期的资源使用特征,通过资源使用特征的阈值在线判断作业终止状态,该方法计算作业特征的方式及预测过程较为简单,不适用于集群环境这类大样本环境。在AFTL的基础上,Tang等<sup>[9]</sup>提出一种基于任务行为模式的异常作业在线预测方法,先分析作业运行过程中资源使用模式,再根据资源使用模式对作业进行聚类,之后对作业聚类中心进行分类。尽管先聚类再分类的方法在一定程度上能提升预测效率,但依然没有解决计算作业特征方式简单这一问题,导致预测效率并未有明显提升。Liu等<sup>[10]</sup>提出的在线序列极限学习机模型(OS-ELM, online sequential extreme learning machine)是一种分阶段预测方法,需要先构建每个时刻作业子任务的资源占用量矩阵,根据此矩阵在线预测作业终止状态。该方法的任务预处理时间较长,且未考虑集群环境下多任务作业的作业与子任务的相关性,导致对作业的预测效率降低且预测耗时增加。

针对上述在线预测方法预测时间长、效率低且未考虑作业与子任务之间状态相关性等问题,笔者首先设计一种新的作业子任务在线阶段的特征计算方式,并根据此特征提出改进门控递归单元(IGRU, improved gated recurrent unit)神经网络,使用IGRU预测子任务运行终止状态。之后进一步通过分析作业与其子任务之间的状态相关性,提出一种新的在线集群异常作业预测(OCAJP, online cluster anomaly job prediction)方法。该方法可在作业运行前期快速、准确预测作业终止状态,提前终止无法正常执行的作业,从而节省集群平台的资源占用,减缓系统调度负担,保障系统的运行安全。

## 1 定义与说明

集群环境下,用户提交作业到集群平台,集群平台将作业分割成ID相同但处理内容不同的多个子任务,并分配至不同的虚拟机等待调度处理。作业子任务的运行过程<sup>[11]</sup>如图1所示。在运行的过程中,若作业子任务出现异常,则系统对此子任务进行反复调度,无异常的子任务正常执行。在作业子任务的运行过程中,由于每一子任务执行状态不同,其具有的特征也不一致<sup>[12-13]</sup>。无法正常执行所有子任务的作业即为执行异常的作业。



根据集群作业处理过程,对于研究中出现的名词做出如下定义。

1) 正常任务:集群环境下成功执行的作业子任务。

2) 异常任务:无法正常完成且可能造成系统负载过大的作业子任务。

由于处理作业过程的特殊性,每一个任务的运行终止状态决定了其所属作业能否正常执行<sup>[4]</sup>,因此属于同一作业的子任务中只要出现一个异常任务,无论其他子任务运行正常与否都会导致作业异常。

3) 状态相关性:子任务的终止状态与其所属作业的终止状态之间的依赖性,即每个子任务运行结果都对其所属作业最终运行结果造成影响。

4) 异常作业:拥有一个或者多个异常子任务的作业。

5) 动态特征:作业子任务在运行阶段每一时刻变化的特征。

对于任务 $i$ 运行阶段中的第 $d$ 个时刻,若此刻资源占用量 $Z_d$ 满足式(1),则该任务的异常次数 $Q_i$ 加1。

$$\left. \begin{aligned} Z_d &> V_{\text{avr}} \\ V_{\text{avr}} &= \frac{\sum_{m=1}^d Z_m}{d}, m = 1, 2, \dots, d \end{aligned} \right\} \quad (1)$$

其中:  $V_{avr}$  为  $d$  时刻之前资源占用量的平均值. 同时, 若第  $d$  个时刻及第  $d-1$  个时刻满足式(2), 则  $Q_i$  累计加 1.

$$\left. \begin{aligned} &|Z_d - Z_{d-1}| > V_{\max} \\ &V_{\max} = \max(|Z_m - Z_{m-1}|), m = 1, 2, \dots, d \end{aligned} \right\} \quad (2)$$

其中  $V_{\max}$  为  $d$  时刻之前资源占用量变动的最大值. 将任务  $i$  在第  $d$  个时刻的  $Z_d, Q_i$  作为其特征值, 并将这些特征值称为任务  $i$  在时刻  $d$  的动态特征.

作业子任务每时刻的特征都会对其终止状态造成影响, 任务执行过程中出现异常频次越高, 则该任务为异常任务的可能性越高<sup>[9]</sup>, 其所属作业为异常作业的概率越大.

## 2 任务在线聚类方法

通过上文对作业子任务在线阶段的动态特征定义和分析可知, 动态特征可用于区分正常任务与异常任务. 下面先依据作业子任务在运行时的动态特征改进普通门控递归单元 (GRU, gated recurrent unit) 神经网络在预测子任务最终运行状态时的不足, 然后说明改进后的 GRU 预测异常任务的流程.

### 2.1 门控递归单元神经网络的改进

在处理具有时间特性的任务动态特征时, GRU 神经网络<sup>[14]</sup>可学习不同时刻的特征信息与任务终止状态之间的依赖关系, 根据每时刻特征实时预测任务将来状态. 为了提高处理效率, GRU 可选择丢弃之前时刻的任务动态特征中的次要信息. 因此在任务运行过程中, GRU 的预测性能明显优于其他机器学习方法<sup>[15]</sup>.

而普通 GRU 采用固定权重处理每时刻变化的任务动态特征, 因此每时刻不同的动态特征信息被丢弃的程度一致, 最终导致处理特征信息的效率降低, 从而影响预测准确度. 为了改进普通 GRU 不能根据任务动态特征动态调节记忆程度, 笔者根据集群任务的动态特征, 使用式(3)中动态权重  $\mu$  代替固定权重.

$$\mu = \varphi \frac{Q_i}{\sum_{i=1}^n Q_i} \frac{Z_i}{\sum_{i=1}^n Z_i}, i = 1, 2, \dots, n \quad (3)$$

其中:  $Q_i$  为当前时刻  $t$  任务  $i$  的异常次数,  $Z_i$  为该时刻任务  $i$  的资源占用量,  $\varphi$  为常数.

改进后的动态权重随着每时刻作业子任务异常次数及使用资源量变化. 在任务执行过程中, 使用动态权重控制特征信息的丢弃程度, 能最大程度保

留异常任务在不同时刻的动态特征.

### 2.2 基于改进门控递归单元的任务在线预测

为了提升在线预测准确度和效率, 基于动态权重和普通 GRU 信息处理过程, 笔者提出一种 IGRU 神经网络, 用于处理任务动态特征并在线预测任务终止状态, 改进了用于衡量信息重要程度的权重计算方式, 使得对动态特征信息的处理过程更符合集群作业的动态特性及集群环境需求. IGRU 神经网络结构如图 2 所示.

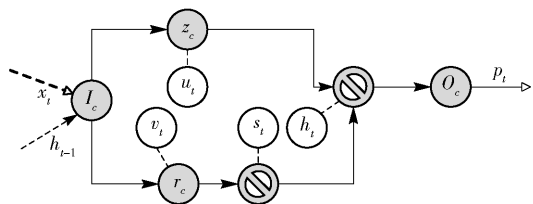


图2 IGRU神经网络结构

IGRU神经网络的处理任务动态特征信息及预测任务终止状态过程设计如下.

**步骤1** 任务  $i$  在时刻  $t$  时, 前一时刻  $t-1$  的输出特征状态信息  $h_{t-1}$  和当前时刻  $t$  的动态特征信息序列  $x_t$  由输入节点  $I_c$  进入 IGRU 神经网络.  $I_c$  将  $h_{t-1}$  及  $x_t$  传送到重置门  $r_c$  与更新门  $z_c$ .

**步骤2** 重置门  $r_c$  决定  $h_{t-1}$  中信息的丢弃程度, 并由式(4)计算信息丢弃之后的状态  $v_t$ .

$$v_t = \mu(x_t W_r + h_{t-1} U_r) \quad (4)$$

其中:  $W_r$  为当前时刻  $t$  到  $r_c$  的连接矩阵,  $U_r$  为上一时刻  $t-1$  到  $r_c$  的连接矩阵. 在时刻  $t$ ,  $v_t$  值越小, 则说明  $h_{t-1}$  中信息被丢弃程度越大.

同时, 更新门  $z_c$  控制  $h_{t-1}$  被保留在当前状态中的程度, 由式(5)得到更新后的状态  $u_t$ .  $u_t$  值越大, 则说明在当前状态中  $h_{t-1}$  及  $x_t$  保留的信息越多.

$$u_t = \mu(x_t W_z + h_{t-1} U_z) \quad (5)$$

其中:  $W_z$  为当前时刻  $t$  到  $z_c$  的连接矩阵,  $U_z$  为时刻  $t-1$  到  $z_c$  的连接矩阵. 由式(3)可知, 动态权重随着动态特征的改变不断变化, 即在每时刻  $t$ , 任务的动态特征决定了  $r_c$  和  $z_c$  对动态特征信息的丢弃及记忆程度.

**步骤3** 使用  $\tanh$  激活函数处理  $v_t$ , 得到待选状态  $s_t$ . 处理后的特征信息  $v_t$  与  $u_t$  以及待选状态  $s_t$  在隐含层融合, 得到丢弃了次要信息, 同时保留重要信息的隐含层状态  $h_t$ .

**步骤4** 由  $h_t$  通过式(6)计算得到当前时刻的最终输出信息  $p_t$ ,  $p_t$  最终通过输出节点  $O_c$  输出为下

一时刻的  $h_{t-1}$ .  $W_o$  表示当前时刻  $t$  到  $O_c$  的连接矩阵,  $\mu$  决定融合后的特征状态信息进入下一时刻的程度.

$$p_t = \mu(h_t W_o) \tag{6}$$

**步骤 5** IGRU 记录时刻  $t$  及对应时刻的最终输出信息  $p_t$ , 同时根据之前每时刻的  $p_t$  与当前时刻特征信息  $x_t$  以及  $p_t$  与  $x_t$  之间的依赖关系预测任务终止状态是否异常.

3 基于改进 GRU 的异常作业预测方法

根据 IGRU 预测任务终止状态过程以及作业相关性, 笔者提出一种 OCAJP 方法的预测流程如图 3 所示.

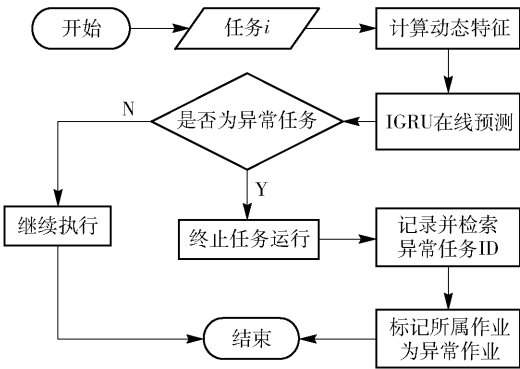


图 3 OCAJP 方法预测流程

IGRU 实时处理任务运行过程中每时刻的动态特征, 并预测作业子任务终止状态是否异常, 之后根据作业的状态相关性, 检索出已被预测为异常任务的所属作业标记为异常作业, 并停止调度或者运行该作业的其他子任务.

OCAJP 方法对异常作业的预测过程设计如下.

**步骤 1** 子任务动态特征计算. 在总量为  $N$  的任务集中, 对于当前时刻  $t$  系统处理的某一任务  $i$ , 系统实时计算  $i$  在该时刻的动态特征, 并生成动态特征序列  $\beta_k^i (k = 1, 2, \dots, k)$ , 得到  $i$  在时刻  $t$  的动态特征数据集  $D_t = \{(i, t, \beta_k^i)\}, i = 1, 2, \dots, N$ .

**步骤 2** 子任务终止状态在线预测. 在当前时刻  $t$ , IGRU 记录并处理  $D_t$  中任务  $i$  在时刻  $t$  之前每个时刻的动态特征序列, 由此实时预测任务终止状态. 在时刻  $t$ , 若 IGRU 预测任务  $i$  的最终状态为异常, 则  $i$  跳转至步骤 3; 若无异常, 则该任务继续运行.

**步骤 3** 在线阶段异常任务处理. 对于在当前时刻  $t$ , 终止状态被预测为异常的子任务, 集群系统

立刻终止此任务的运行并记录其任务 ID, 然后执行步骤 4.

**步骤 4** 在线阶段异常作业检索. 实时检索与步骤 3 中异常任务 ID 一致的任务并停止其运行, 同时将该任务所属作业标记为异常作业, 结束作业预测.

4 实验与结果

4.1 实验数据及环境配置

在验证实验中, 选取原始数据集<sup>[1]</sup>中前 10 天的作业数据集. 使用 Python 中的 Pandas 库对前 10 天的作业数据集进行筛选及清洗, 去除掉记录不完整的数据包括在记录之前就已经运行的作业、在记录结束后还未结束的作业、未调度已经结束的作业等有缺失信息的作业, 清洗之后得到的数据集包含 18 623 个作业.

筛选后的作业包含了预测需要的所有特征值信息及最终的运行状态, 随机选取 1.8 万个作业作为实验数据集, 并且采用笔者对正常作业及异常作业的定义为实验数据集中作业进行分类. 从实验数据集中随机抽取作业作为训练集和测试集, 分别用于方法训练及预测效果测试, 实验环境配置见表 1.

表 1 实验环境配置

硬件配置	软件环境
Intel Core i5-3570 CPU@ 3.40 GHz	Windows 7 64 位操作系统
4 GB 内存, 500 GB 磁盘	MySQL, Python, Matlab

4.2 评价指标与实验过程

4.2.1 评价指标

在相同实验环境下, 选取普通 GRU、AFTL 算法、OS-ELM 方法和本文的 OCAJP 方法, 对实验数据集中的作业分别进行预测. 为对比上述 4 种方法的预测性能, 定义灵敏度、误差率、精确度和预测时长 4 个评价指标.

1) 灵敏度: 被正确预测为异常作业占真实异常作业总量的比例.

2) 误差率: 被错误预测为异常作业的真实正常作业数量占被预测为异常作业总数的比例.

3) 精确度: 预测结果与实际结果一致的作业数在总作业数中的占比.

4) 预测时长: 预测方法正确预测作业终止状态时, 作业已经在云集群系统中的运行时间.



### 4.2.2 实验过程

在实验中,选取 CPU 占用量、RAM 及 DISK 的占用量作为动态特征中的系统资源占用量属性. 实验计算上述各方法的预测灵敏度、误差率、精确度和预测时长,计算多次预测结果的平均值并进行对比. 实验过程设计如下.

**步骤1** 划分训练集. 从实验数据集中随机抽取 9 000 个作业作为训练集  $S_\alpha$ . 不使用全部实验数据集训练方法,能保证之后测试实验结果的有效性.

**步骤2** 方法训练. 为使实验结果更准确,同时之前并无使用普通 GRU 方法预测集群异常作业的研究,需先对 OCAJP 方法及普通 GRU 方法进行训练. 输入训练集  $S_\alpha$ , 训练集中任务  $i$  的真实终止状态表示为  $y_i$ , 方法对  $i$  的预测结果为  $y'_i$ . 预测结果与真实终止状态之间的误差表示为

$$E = \begin{cases} 1, & y_i = y'_i \\ 0, & y_i \neq y'_i \end{cases} \quad (7)$$

当  $y_i = y'_i$  时,表示预测结果与任务实际终止状态一致,此时  $E = 1$ ; 若  $E = 0$ ,则表示预测结果与任务实际终止状态不符,将  $E = 0$  时的任务重新输入,重复训练过程.

**步骤3** 测试集划分. 每次实验时,从实验数据集中随机抽取 0.3、0.6、0.9、1.2、1.5、1.8 个作业分别作为测试集  $S_t$  ( $t = 1, 2, 3, 4, 5$ ),用于验证各方法的预测效果.

**步骤4** 方法测试与结果处理. 输入测试集  $S_t$ , 分别对 GRU、AFTL 算法、OS-ELM 方法及 OCAJP 方法进行测试,得到各方法的预测结果. 计算每种方法的预测效率及预测时间,记录数据并重复实验步骤,每次实验均重新抽取测试集,最终计算得到 10 次实验结果的平均值.

## 4.3 实验结果与分析

### 4.3.1 灵敏度

OCAJP 方法与普通 GRU、AFTL 算法、OS-ELM 方法的作业预测灵敏度如图 4 所示. 由图 4 可见, OCAJP 与 OS-ELM 方法的预测灵敏度明显高于 GRU 及 AFTL 方法.

随着作业数量的增加,GRU 及 AFTL 方法的预测灵敏度均下降得较为明显,其中 AFTL 方法的预测灵敏度几乎呈直线下降,说明普通 GRU 及 AFTL 方法在预测大规模样本时处理异常作业特征效率较低. 在线预测过程中,OCAJP 方法通过实时计算资源占用量所得到的任务动态特征能更准确地表示任

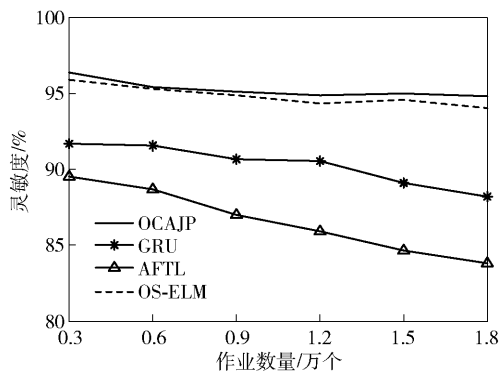


图4 预测灵敏度对比

务的当前执行状态,并且改进后的 IGRU 神经网络在处理动态特征时效率更高,因此 OCAJP 方法的预测灵敏度高于其他方法.

### 4.3.2 误差率

对 OCAJP 方法与 GRU、AFTL 方法、OS-ELM 方法的预测误差率进行测试,重复实验并计算 10 次预测灵敏度的平均值,最终结果如图 5 所示.

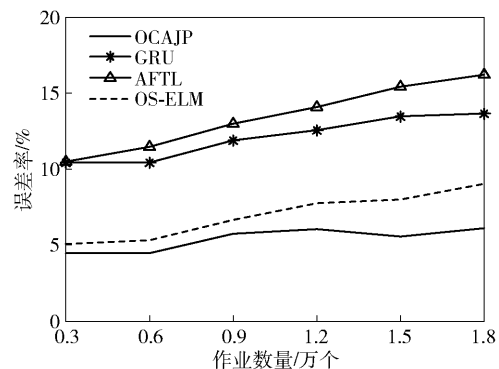


图5 预测误差率对比

由图 5 可见,普通 GRU 及 AFTL 方法在预测异常作业的误差率方面表现较差;OS-ELM 方法在预测初期即预测作业数量较小时预测误差率较小,而随着预测作业数量的增加,OS-ELM 方法的预测误差率几乎呈直线上升. 而 OCAJP 方法的预测误差率明显低于其他几种方法,并且在作业数量较大时,其预测误差率增长幅度较小. 因为 OCAJP 方法先预测作业子任务终止状态,之后根据这一结果及作业相关性预测异常作业,而不使用作业子任务特征作为作业特征直接对作业进行预测,使得预测误差较小.

### 4.3.3 精确度

经过多次实验并计算 OCAJP 方法与 GRU、AFTL 算法和 OS-ELM 方法的作业预测精确度平均值,

对比结果如图6所示。OCAJP方法及OS-ELM方法的预测精确度表现显著,优于普通GRU及AFTL算法。

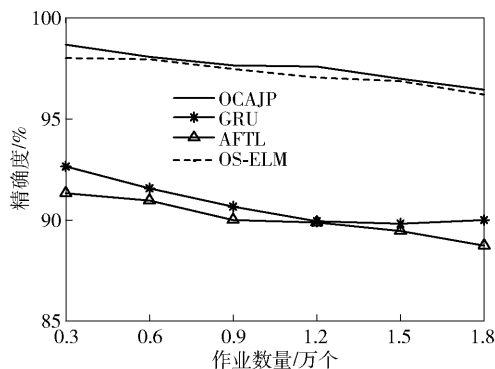


图6 预测精确度对比

尽管OCAJP方法与OS-ELM方法在预测精确度上的差距不明显,但OCAJP方法的预测精确度依然高于OS-ELM方法。并且随着作业数量的增长,OCAJP方法的预测精准度尽管有所下降,但即使在预测的作业数量较大时,OCAJP方法的预测精准度依然超过其他几种方法。OCAJP方法在线预测时,IGRU由动态权重控制动态特征信息丢弃程度的方式使得预测结果更精准;此外,OCAJP方法根据作业与任务之间的相关性检索异常作业的方式能准确标记异常作业,从而提高预测异常作业的精确度。说明OCAJP方法的预测过程比较符合集群动态性,并且能高效处理集群作业的动态特征。

#### 4.3.4 预测时长

预测时长不仅能代表方法预测作业的性能,同时能反映各预测方法节省系统资源的能力。上述4种方法在10次实验中的平均预测时长如图7所示。由图7可见,AFTL算法及普通GRU的预测时长随着作业数量的增加几乎呈线性增长,说明AFTL算法及普通GRU不适用于大样本预测,并且对作业的预测效率较低;采用IGRU神经网络实时预测任务终止状态使得该方法能更好地处理任务动态特征,从而显著减少普通GRU所需预测时长。

尽管OS-ELM方法与OCAJP方法的预测时长差距较小,但OCAJP方法通过作业与其子任务之间相关性检索异常作业,避免了对同属于相同作业的不同异常任务进行重复预测,可以有效降低预测时长,因此OCAJP方法的预测时长少于OS-ELM方法。图7的实验结果表明,在预测大规模作业数据时,OCAJP方法能及时预测作业异常,可有效缩短

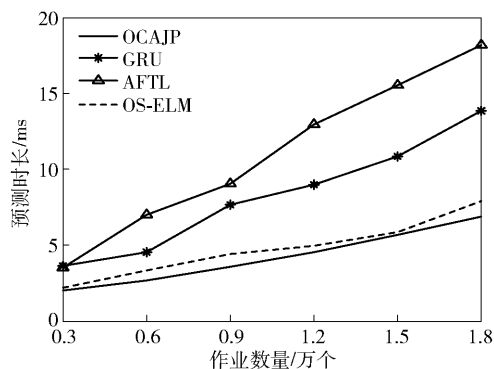


图7 预测时长对比

异常作业在云集群系统中的运行时间,从而及时预测出异常作业,减小异常作业对系统资源的消耗,保障集群系统安全。

## 5 结束语

为克服现有异常作业在线预测方法预测效率低、预测耗时长且未考虑作业相关性等不足,通过分析集群作业处理特性,提出从作业子任务异常角度考虑作业的异常。首先设计动态特征计算方式;其次根据这一动态特征设计动态权重用于弥补普通GRU不能动态调节记忆程度这一不足;之后基于作业子任务的动态特征和改进后的IGRU以及作业相关性,提出一种基于IGRU的异常作业在线预测方法,先预测作业子任务的运行终止状态,再根据作业相关性检索异常作业;最后通过对比实验验证说明OCAJP方法的预测性能明显优于其他方法,即OCAJP方法能快速、高效地预测出集群异常作业,减小异常作业在系统中运行造成的资源消耗,从而保障集群平台的安全性。

因集群作业数目巨大,同时在线预测所有作业会增大系统预测压力,未来考虑进一步分析作业子任务在离线阶段所具备的特征,并根据此特征增加离线时对作业的预测,从而减弱在线预测时的预测压力,最终进一步提升作业预测的效率,并缩短预测耗时。

## 参考文献:

- [1] Google cluster data [EB/OL]. (2010-01-10) [2018-09-10]. <http://googleresearch.blogspot.com/2010/01/google-cluster-data.html>.
- [2] 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.

- 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.
- [3] 王意洁, 孙伟东, 周松, 等. 云计算环境下的分布存储关键技术[J]. 软件学报, 2012, 23(4): 962-986.
- Wang Yijie, Sun Weidong, Zhou Song, et al. Key technologies of distributed storage for cloud computing [J]. Journal of Software, 2012, 23(4): 962-986.
- [4] Soualhia M, Khomh F, Tahar S. Predicting scheduling failures in the cloud: a case study with Google clusters and Hadoop on Amazon EMR [C] // 2015 IEEE 17<sup>th</sup> International Conference on High Performance Computing and Communications. Piscataway: IEEE Press, 2015: 58-65.
- [5] Chen X, Lu C D, Pattabiraman K. Failure analysis of jobs in compute clouds: a Google cluster case study [C] // 2014 IEEE 25<sup>th</sup> International Symposium on Software Reliability Engineering. Piscataway: IEEE Press, 2014: 167-177.
- [6] Jakobik, Agnieszka, Grzonka D, Palmieri F. Non-deterministic security driven meta scheduler for distributed cloud organizations [J]. Simulation Modelling Practice and Theory, 2017, 76(8): 67-81.
- [7] Sonoda M, Kikuchi S, Watanabe Y, et al. Online failure prediction in cloud datacenters by real-time message pattern learning [C] // Proceedings of the 2012 IEEE 4<sup>th</sup> International Conference on Cloud Computing Technology and Science Proceedings. Piscataway: IEEE Press, 2012: 504-511.
- [8] 唐红艳, 李影, 贾统, 等. 基于时间序列分析的杀手级任务在线识别方法 [J]. 计算机科学, 2017, 44(4): 43-46.
- Tang Hongyan, Li Ying, Jia Tong, et al. Time series based killer task online recognition approach [J]. Computer Science, 2017, 44(4): 43-46.
- [9] Tang H, Li Y, Jia T, et al. Hunting killer tasks for cloud system through behavior pattern learning [C] // IEEE/IFIP International Conference on Dependable Systems & Networks Workshop. Piscataway: IEEE Press, 2016: 1-12.
- [10] Liu C, Han J, Shang Y, et al. Predicting of job failure in compute cloud based on online extreme learning machine: a comparative study [J]. IEEE Access, 2017, 5(99): 9359-9368.
- [11] Garraghan P, Townend P, Xu J. An empirical failure-analysis of a large-scale cloud computing environment [C] // 2014 IEEE 15<sup>th</sup> International Symposium on High-Assurance Systems Engineering. Piscataway: IEEE Press, 2014: 113-120.
- [12] Rosa A, Chen L Y, Binder W. Predicting and mitigating jobs failures in big data clusters [C] // 2015 15<sup>th</sup> IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. Piscataway: IEEE Press, 2015: 221-230.
- [13] Yamnual K, Phunchongharn P, Achalakul T. Failure detection through monitoring of the scientific distributed system [C] // 2017 International Conference on Applied System Innovation. Piscataway: IEEE Press, 2017: 568-571.
- [14] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation [J]. Computer Science, 2014, 55(9): 1406-1420.
- [15] Lipton Z C, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning [J]. Computer Science, 2015, 56(10): 1506-1543.