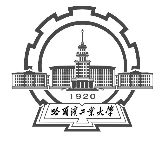
哈尔滨工业大学

**<<大数据分析>>**

**课程实验报告**

**(2024年度秋季学期)**

|  |  |
| --- | --- |
| **姓名：** | **郁清方** |
| **学号：** | **2022110900** |
| **学院：** | **计算机技术与科学** |
| **教师：** | **杨东华** |



Harbin Institute of Technology

**大数据分析课程报告**

课程名称： 大数据分析

报告类型： 课程报告

报告题目： 基于机器学习的系统日志异常检测系统

所在院系： 计算学部

所在学科： 数据科学与大数据技术

学生类别： 4年制本科生

任课教师： 杨东华

学生姓名： 郁清方

学生学号： **2022110900**

选课时间： 2024秋

评阅成绩：

哈尔滨工业大学

基于机器学习的系统日志异常检测系统

**摘要：** 本文构建了一个使用LDA主题模型以及LSTM神经网络的日志序列异常检测系统，通过学习无异常的日志序列，在推理时寻找出异于正常的日志序列，从而达到日志序列异常检测的目的。在推理部分基于MapReduce框架实现LDA和LSTM的推理过程。然后展示了相关算法的伪代码实现。最后提出了模型的评估方法和评估步骤。

**关键词： 时间序列的异常检测模型、Hadoop、MapReduce、LDA主题模型、神经网络LSTM**

# 第1章 需求分析

## 1.1 研究问题的背景

随着网络和新一代通讯技术的高速发展，在线服务需求激增，数据成为关键资源，而数据中心承担着数据收集、计算、转化、流通的重要职责，日常运行负载始终处于较高状态，加之数据中心通常由成百上千台服务器组成集群，而且随着软件系统越来越庞大和复杂，一个数据中心的各个软硬件模块每日产生的系统日志数是海量的，运维难度与成本也显著增高。

以 HDFS(Hadoop Distributed File System)分布式文件系统为例，该系统可在 38.7 小时的运行时间内，产生超过11,175,000 条日志记录，并且日志文件的体积可达 1.47GB 。显然在目前的趋势下，单单依靠人工标记分析海量系统日志已经变得不太可能，迫切需要一种大数据分析的应用系统来辅助运维人员的工作，有利于故障提前预警和故障发生后的快速定位。

## 1.2 研究问题的挑战

经过分析，我认为主要有以下三个方面的挑战：

1. **海量日志的处理的挑战**：

庞大的数据中心往往有着多个节点，每个节点的相关信息混杂在日志文件当中。

1. **模型随时间变化出现精度下降的挑战**：

随着系统的不断升级，新的功能也在不断产生，日志的内容和结构也会随之更新，而模型无法准确识别新的日志，使得系统发现异常的精度随之下降。

1. **数据集污染挑战**：

训练所使用的数据集，大多是由互联网公司提供，它们的服务器可能会遭受攻击，这导致日志数据可能存在篡改的可能。

## 1.3 本文解决问题的方法

本系统采用机器学习和深度学习相结合的方法，借助已有的日志解析器对日志进行模版提取，基于MapReduce框架构建 LSTM 的日志主题预测模型，利用相关算法将系统日志模板化、提取特征并随后进行异常检测，从而实现从系统日志中筛选出可能存在异常的日志。

**原理：通过学习无异常的日志序列，学习正常的系统时序行为，通过寻找与预测结果相异的真实序列，从而实现异常日志检测的快速定位。**

整个日志异常检测系统的模块如下：



图1：日志异常检测系统的模块

训练以及测试部分的总体流程（考虑到在MapReduce框架上实现机器学习和深度学习的训练存在效率低下的问题，训练部分我选择不使用MapReduce框架而选择在python环境下，借助pytorch、sklearn直接使用CPU或者GPU进行训练，但是验证/测试部分将基于MapReduce实现）：

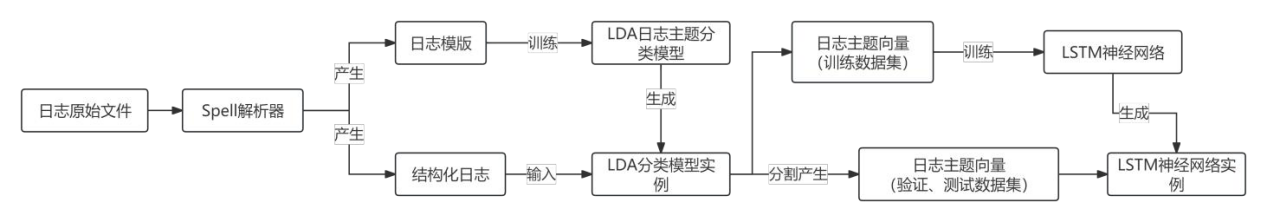


图2：训练以及测试部分的总体流程

# 第2章 系统设计

## 2.1 整体架构

系统的整体架构如下：



图3：系统整体架构

## 2.2 数据集说明

系统的日志有很多种类，在这里我选用了HDFS的日志作为数据集，数据来自 Github（https://github.com/logpai/loghub）。该数据集采集于 200 多台亚马逊 EC2 节点，其中涵盖了 575,061 个 block节点所产生的 11,175,629 条日志记录，其中有 16,838 个 block 节点被 Hadoop 数据领域的专家标记为异常。

在训练模型的时候，将会提取其中的无异常的blockId的序列进行训练。

## 2.3 威胁模型

### **2.1 引入原因**

在本系统中，我所提出的方法将会从无异常的日志序列中学习到全面而复杂的进程行为模式。

由于训练所用到的日志数据来自互联网巨头的共享，难免存在服务器被恶意攻击而产生的日志，针对这个问题，我在这里引入两种威胁模型。

假设系统产生的日志记录不会被敌对攻击者所篡改，并且敌对攻击者也不能通过篡改系统代码从而使系统无法正常地产生日志记录。因此，假设系统日志服务是安全可信的。系统日志服务能够将异常进程行为或者敌对攻击者的行为记录到日志文件内。

### **2.2 引入的两种威胁模型**

(1) 敌对攻击导致的系统行为模式产生异常。此类威胁通常是敌对攻击针对系统，从而导致系统产生异常的日志行为模式、服务不断重启以及某些任务由于异常出现而提前中断。

(2) 敌对攻击在系统日志中所留下的痕迹。此类威胁通常是敌对攻击针对系统环境下的某些子任务进程，从而在系统运行过程中留下自己相应的日志痕迹。

## 2.4 三条主要数据流

在本应用系统当中，有三条主要的数据流，分别是：

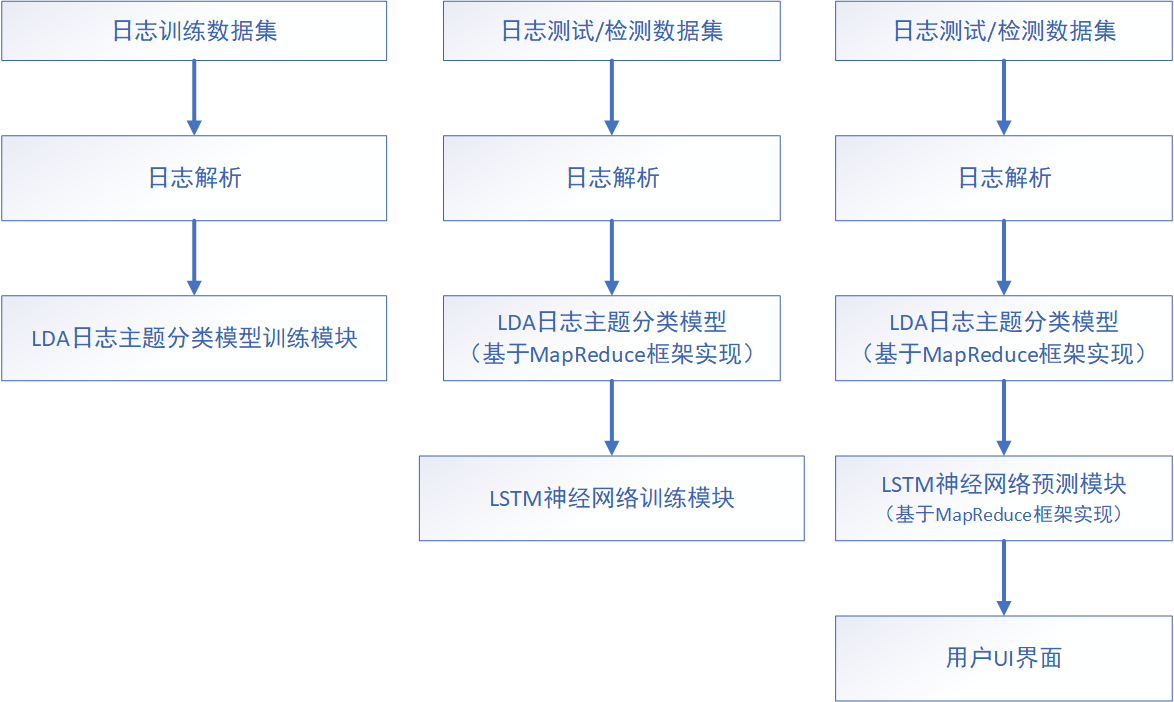


图4：本系统的三条主要数据流

前两条数据流用于训练整个系统，最后一条数据流用于日志的异常检测。

对于常规的日志异常检测，主要涉及以下内容：日志采集、日志解析、特征提取以及异常检测。本系统主要聚焦于后面的三个步骤。

## 2.5 日志解析

日志属于半结构化数据，需要分析其一般构成，并利用解析技术将其中的常量和变量分离开来，为后续的特征提取提供良好的基础。

一条日志数据可以分解为正则消息部分和特征消息部分。正则消息包括时间戳、日志等级、 产生日志的类名称等日志基本组成这部分是可以用正则表达式进行提取和分解的，属于日志数据的基础信息。特征消息部分则是描述了日志内容的核心部分，由文本、数字及特殊符号等组成。

本模块通过日志文本，对日志进行解析，并将日志记录转化为结构化日志，即将日志拆分为日志模板和参数变量。

由于我选择了 HDFS 的日志作为本系统的数据集，所以我选择了针对 HDFS 的日志解析效果最好的为Spell 解析器（在这里我使用github上https://github.com/logpai项目实现的spell解析器，可通过安装python包：pip install logparser3，调用实现，算法部分就不讲实现了）。

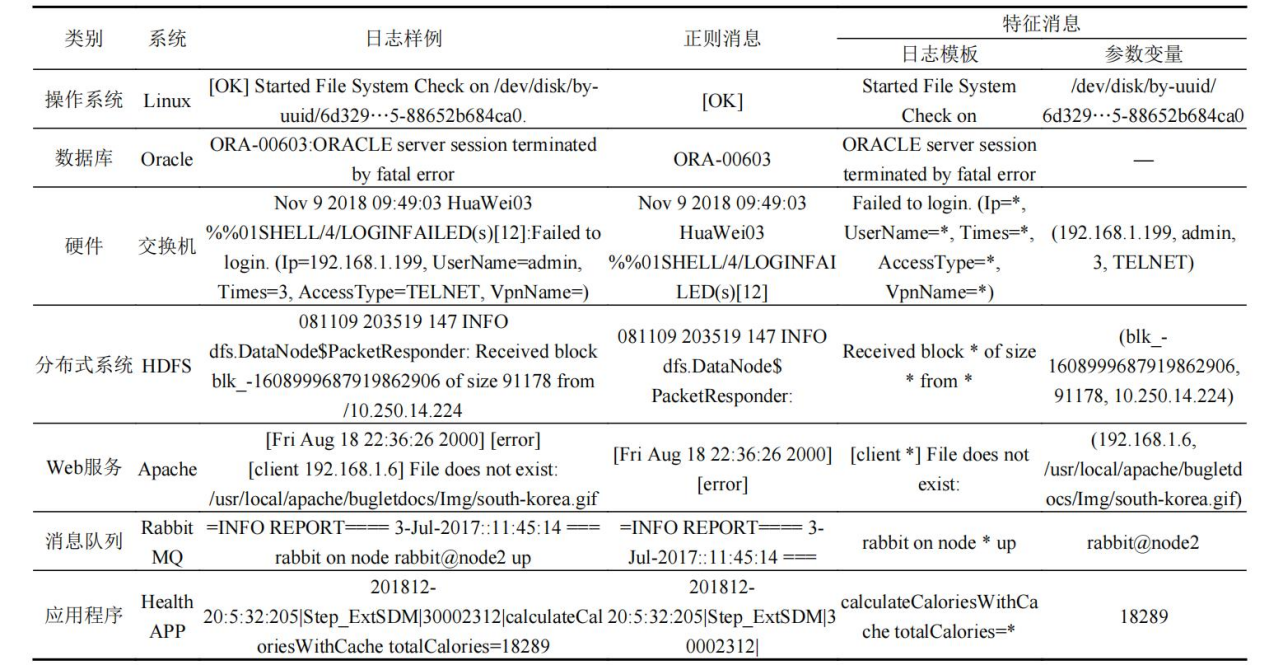


图5：日志解析样例

## 2.6 特征提取

本模块主要在于构造机器学习模型可以处理的特征数据，从而来学习日志的正常或异常模式。所提取的特征质量决定了后期模型检测效果所能达到的精度。

查阅相关文献和技术资料得知，特征提取往往需要首先将日志拆分成不同的小组，每个小组为异常检测的最小单位，而拆分方式主要分为固定窗口(Fixed Window)、滑动窗口(Sliding Window)和会话窗口(Session Window)。

在本系统之中，我们采用了滑动窗口(Sliding Window)的方式。

滑动窗口(Sliding Window)的示例如下：

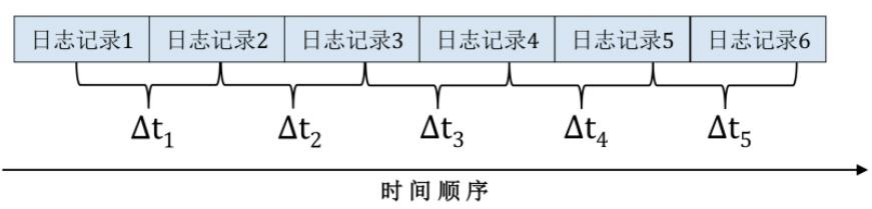


图6：滑动步长为 1，窗口长度为 2，日志总长为 6 的示例

通过构建一个基于 LDA 主题模型的日志模版分类模型来将结构化的日志转化为特征向量，将此特征向量输入到后续的异常检测模块之中。

这个日志模版分类模型含有两个部分：映射字典和LDA主题模型，训练时将使用日志模版作为语料库训练LDA主题模型；匹配时将日志模版输入主题分类模型并输出对应的主日志模版主题（这是一个特征向量，采用one-hot独热码方式编码），具体算法将在算法演示部分讲解。

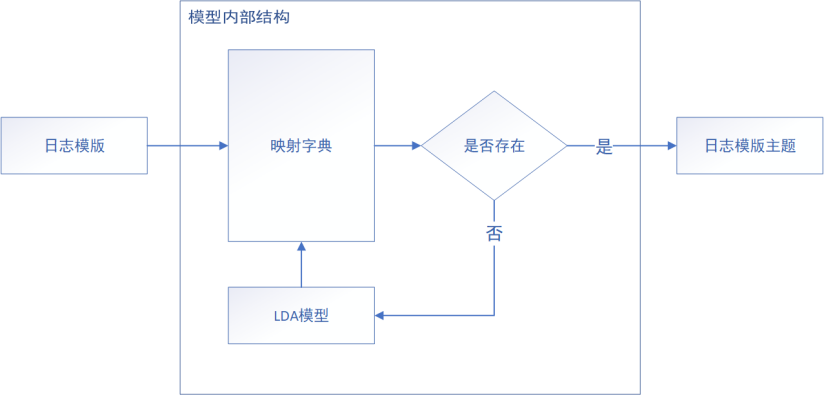


图7：日志模版分类模型内部结构

同时，在日志模版分类模型输出时，会以滑动窗口的形式（在本系统当中我设置滑动窗口大小为6，步长为1）分组输出日志，方便输入后续的LSTM神经网络对其进行操作：

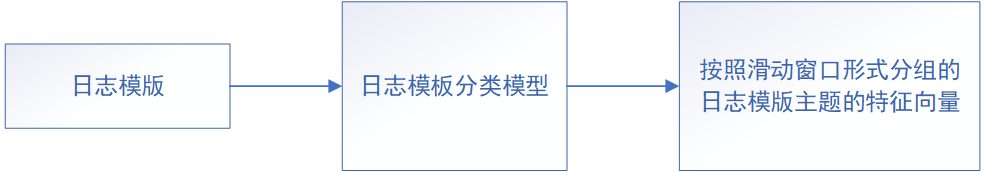


图8：日志模板分类模型的输入输出

## 2.7 异常检测

经过特征提取，原始日志数据已经转换为模型可以处理的特征数据，可以输入判别模型进行异常检测。

在查阅多篇文献后，我决定采用 LSTM（Long Short Term Memory）循环神经网络。LSTM 是循环神经网络的一种改进模型，相较于传统的 RNN(Recurrent Neural Network)循环神经网络， LSTM 通过将记忆状态拆分为长期记忆和短期记忆，从而解决了传统 RNN 容易丢失长期记忆信息的缺陷问题。

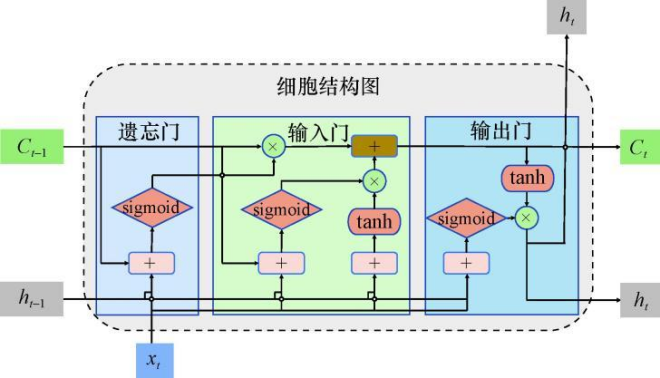


图9：LSTM的细胞结构

在本应用系统中，我搭建了一个具有 3 个全连接层，每层含有 5 个 LSTM 神经元，每个 神经元的输入为日志模版主题的独热码，输出层使用 softmax 函数转化为日志模版主题的概率分布来预测潜在的下一个日志模版主题。

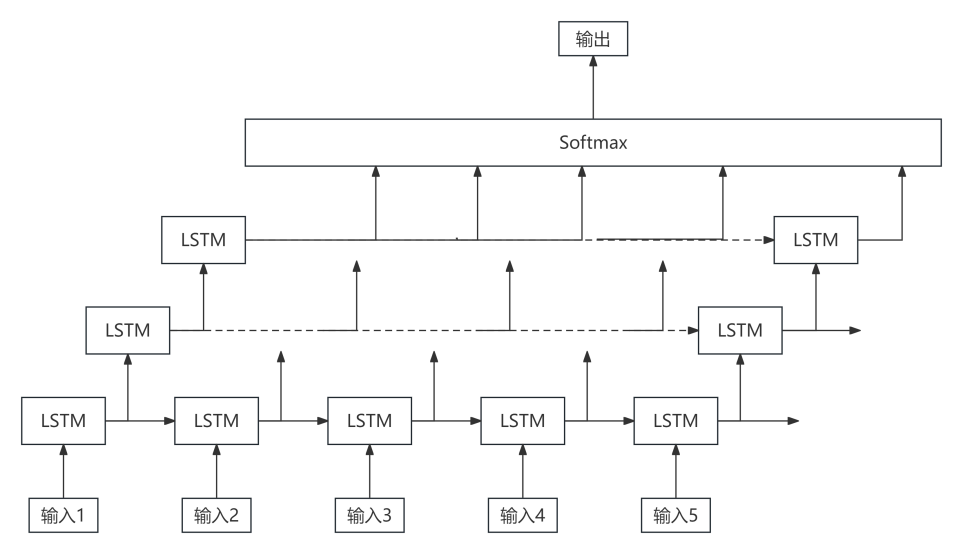


图10：LSTM神经网络结构

**判断日志是否存在异常的规则：**LSTM通过读取来自上层日志模板主题模型输出的分组日志（上面提及窗口大小为6，步长为1），将前5个特征向量输入到LSTM神经网络当中，输出下一个日志模板主题概率分布，挑选前K（K在此处选定为日志模板总数的20%）个特征向量，如果输入的分组日志的最后1个特征向量在这K个特征向量当中，那么认为没有检出异常，否则认为出现异常。

## 2.8 用户UI界面

用户UI界面主要负责调用以上几个模块执行并显示各模块执行完毕后的结果。

本应用系统还支持增量更新功能：可以定期进行训练以满足系统日志不断更新的需求。

LSTM检测出的疑似异常日志将会发送给用户UI界面并展示，同时，如果人工在复核后发现有误报的情况，那么会将误报的信息添加进训练集当中，当新的训练集达到一定规模之后，执行增量更新。

## 2.8 算法演示(含代码或伪代码)

在本部分当中，主要实现以下算法的伪代码：

日志模板分类模型训练算法（LTCMTA，Log Template Classification Model Train Algorithm）

日志模版分类模型匹配算法（LTCMMA，Log Template Classification Model Match Algorithm，基于Hadoop框架实现）

日志异常检测模型训练算法（LADMTA， Log Anomaly Detection Model Train Algorithm）

日志异常检测模型检测算法（LADMDA， Log Anomaly Detection Model Detection

Algorithms，基于Hadoop框架实现）

### 2.8.1 日志模板分类模型训练算法LTCMTA

|  |
| --- |
| **算法：LTCMTA** |
| 输入：日志模版，人为指定主题数量范围集合K  输出：LDA主题模型实例tcm，最佳主题数量   1. 初始化，导入必要的库和模块 2. 定义停用词集合：使用nltk库的stopwords获取英文停用词 3. 使用pandas的read\_csv读取日志模版文件到DATAFrame 4. 文本预处理：对DataFrame中的'EventTemplate'列进行预处理（大小写转换、分词、去除停用词和非字母字符） 5. FOR EACH k IN K DO 6. 创建字典和语料库，指定lda主题数量k 7. 训练lda模型 8. END FOR; 9. 计算不同主题数量的一致性得分 10. 选择当前一致性得分最高的lda模型，导出实例tcm 11. RETURN最佳主题数量 |

### 2.8.2 日志模版分类模型匹配算法LTCMMA

本算法将基于MapReduce框架实现，Mapper负责按照不同的blockId，将属于不同的blockId写入不同的Reducer，Reducer获取先前训练好的lda实例，随后读取结构化日志随后输出日志主题的特征向量，并以窗口大小为6，步长为1的滑动窗口方式为一组，输出训练集/测试集。

|  |
| --- |
| **算法：LTCMMA\_Job** |
| 输入：输入文件路径，输出文件路径  输出：无   1. 检查输入输出路径是否合法 2. 配置MapReduce任务：输入输出路径，Mapper和Reducer类 3. 布置MapReduce作业 4. 等待MapReduce作业完成 |

|  |
| --- |
| **算法：LTCMMA\_Mapper** |
| 输入：结构化日志序列  输出：区分了blockId的日志序列（MapReduce特性）   1. 读取结构化日志序列 2. 提取其中的blockId 3. 按照键值对<blockId, 日志序列>向Reducer写入数据。 |

|  |
| --- |
| **算法：LTCMMA\_Reducer** |
| 输入：区分了blockId的日志序列（来自Mapper），LDA主题模型实例lda  输出：<blockId, 按照滑动窗口形式分组的日志序列的特征向量>   1. 读取并装载LDA模型实例lda，设置映射字典，初始化为空 2. FOR EACH value IN Value DO 3. 查询映射字典 4. IF 没有找到 THEN 5. 向lda输入该日志序列，并将输出结果添加到映射字典当中 6. END IF; 7. 将输出结果（特征向量）存入一个队列q当中 8. IF q.size() == 6 THEN 9. 将这6个特征向量组合成一组，按照<blockId, 6个特征向量的分组>写入输出 10. 将q的头删去1个 11. END IF 12. END FOR |

### 2.8.2 日志异常检测模型训练算法LADMTA

|  |
| --- |
| **算法：LADMTA** |
| 输入：主题模版分类模型输出的训练集数据  输出：LSTM模型实例lstm   1. 初始化，导入必要的库和模块 2. 定义LSTM神经网络 3. 设置损失函数和优化器（我使用的是交叉熵损失和adam优化器） 4. 分批次训练，指定epoch大小为1000，学习率为0.001，batch大小为512 5. 训练完毕，输出LSTM模型实例lstm |

### 2.8.2 日志异常检测模型检测算法LADMDA

本算法将基于MapReduce实现，Mapper负责按照不同的blockId，将属于不同的blockId写入不同的Reducer，Reducer负责读入LSTM实例lstm，进行日志异常检测，最后将异常的blockId和对应的日志序列输出

|  |
| --- |
| **算法：LADMDA\_Job** |
| 输入：输入文件路径，输出文件路径  输出：无   1. 检查输入输出路径是否合法 2. 配置MapReduce任务：输入输出路径，Mapper和Reducer类 3. 布置MapReduce作业 4. 等待MapReduce作业完成 |

|  |
| --- |
| **算法：LADMDA\_Mapper** |
| 输入：来自日志模版分类模型的输出  输出：<bolckId, 日志序列分组>   1. 获取blockId 2. 按照键值对<bolckId, 日志序列分组>向Reducer写入 |

|  |
| --- |
| **算法：LADMDA\_Reducer** |
| 输入：键值对<bolckId, 日志序列分组>（来自Mapper）  输出：认为存在异常的blockId和对应的日志序列   1. 初始化，读取LSTM实例lstm 2. FOR EACH 分组g IN 日志序列分组 DO 3. 提取出分组g的前5个特征向量输入lstm模型 4. 计算最终softmax层的输出 5. 选择前K个概率最大的生成对应的特征向量，形成集合A 6. IF g的第6个特征向量不在集合A当中 THEN 7. 将对应的blockId按照键值对<bolckId, 日志序列分组>写入输出当中 8. END IF 9. END FOR |

# 第3章 结果评估

## 3.1 说明

在上文提到的数据集当中，既包含了正常的blockId的日志序列，也包含了异常的blockId的日志序列，在训练时主要只使用了正常的blockId。

所以在结果评估的时候，将会使用正常的blockId和异常的blockId的混合日志序列作为原始输入，让本系统进行异常检测，根据输出的异常blockId以及预先知道的异常blockId、blockId总数进行评估。

**几个定义：**

1. 误检数（False Positive）：在所有被检测的blockId中，被错误地判定为异常的blockId数量，记为 FP。
2. 漏检数（False Negative）：在所有被检测的blockId中，被错误地判定为正常的blockId数量，记为 FN。
3. 查准率（Precision）：正确检测出的异常的blockId的数量，占所有检测为异常的blockId的比重，用 TP 表示正确检测出的异常blockId的数量，计算公式如下：
4. 查全率（Recall）：正确检测的异常blockId的数量，占所有确定异常的blockId数量的比重，计算公式如下：
5. 调和分数（F1 Score）：精确率和召回率的调和均值，记为 FS，计算公式如下：

## 3.2 模型评估

模型评估的流程如下：

|  |
| --- |
| **基于机器学习的系统日志异常检测模型评估流程** |
| 1. 将含有异常和无异常blockId的日志序列混合，统计其中异常和无异常的blockId的数量 2. 将步骤1当中混合的数据输入系统当中 3. 获取系统的输出结果，主要关注输出的blockId 4. 根据输入数据前统计的blockId的真实情况，计算输出结果当中的TP、FP和FN 5. 根据TP、FP和FN的值进一步计算Precision、Recall以及F1 Score |

获得的Precision、Recall可以显示本系统在当测试集上检测异常的blockId的异常谨慎度和异常敏感度

也可将获得的Precision、Recall和F1 Score与其他现有的日志异常检测模型，如Deeplog、LogAnomaly，在相同数据集下进行测试，进行比对。

## 3.3 后记

本大作业的内容和灵感来自于我的大二年度项目，只是当时做到后面两个组员摆烂了，迫不得已全部自己手搓全栈，做了个简易的前后端，把整个项目串联成为了一个应用。

其结题报告见[附件1：11\_2023F0145\_基于机器学习的系统日志异常检测\_郁清方.pdf]