

分 类 号 \_\_\_\_\_  
学校代码 10487

学号 M201672677  
密级 \_\_\_\_\_

# 华中科技大学

# 硕士学位论文

## 大规模磁盘故障预测方法研究

学位申请人： 谢伟睿

学 科 专 业： 计算机系统结构

指 导 教 师： 李春花 副教授

答 辩 日 期： 2019 年 5 月 29 日

**A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree for the Master of Engineering**

**Research on Large-scale Disk Failure  
Prediction Method**

**Candidate : Xie Weirui**

**Major : Computer Architecture**

**Supervisor : Assoc. Prof. Li Chunhua**

**Huazhong University of Science and Technology**

**Wuhan, Hubei 430074, P. R. China**

**May, 2019**

## 独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：

日期：          年          月          日

## 学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 ☐ ，在 \_\_\_\_\_ 年解密后适用本授权书。

本论文属于

不保密 ☐ 。

（请在以上方框内打“√”）

学位论文作者签名：

指导教师签名：

日期：          年          月          日

日期：          年          月          日

## 摘要

大规模存储系统中磁盘故障预测至关重要，一旦磁盘损坏信息丢失，将会给企业带来无法挽回的损失。基于磁盘运行数据，使用机器学习方法进行磁盘故障预测，目前已可得到较好的预测效果。但由于 S.M.A.R.T 属性对故障表征能力的局限性，仅含单个时间点的信息导致预测效果不佳。在磁盘故障预测问题中，目前预测的召回率和误报率无法达到工业级应用水平。

文章将时序信息引入磁盘预测模型，提出时序随机森林的磁盘故障预测方法，包括数据时序特征处理和时序模型的优化。时序特征处理部分，提出数据时序分配策略，将时序数据按照时间动态分配到训练集、验证集与测试集；提出时序特征处理算法，将 S.M.A.R.T 属性的变化值拓展到模型特征中。预测模型优化部分，提出随机森林决策树时序权重投票算法，基于样本数据的时序特征，对随机森林中的不同决策树初始化不同权重，使用均值和方差进行时序数据的拟合；提出负反馈更新模型，在随机森林中引入后验决策树；基于历史预测数据加入负反馈信息，提出决策树权重更新算法。基于上述研究，设计并开发了数据中心的磁盘故障预测原型系统，包括磁盘数据采集子系统、磁盘故障预测子系统、磁盘巡检子系统与中心监控子系统，现已部署在大规模数据中心。

通过测试与分析，结果表明文章提出的时序预测模型优于传统机器学习模型。相比于传统模型，使用时序数据模型后，在同样的误报率下召回率提高 11.13%；使用优化的时序模型后，在同样的召回率下误报率降低 52.0%；两者一起使用时召回率与误报率皆有稳定提升。将预测结果应用于磁盘巡检后，加速巡检时间占比为 5.0% 时，平均故障探测时间降低了 152.6%；磁盘的磁盘巡检负载增加 4.8% 时，平均故障探测时间降低了 217.3%，可大幅度减少磁盘巡检开销，降低平均故障探测时间。

**关键词：**磁盘故障预测、S. M. A. R. T 技术、磁盘巡检、决策树

## Abstract

Disk failure prediction in large-scale storage systems is critical, and once disk corruption information is lost, it will bring irreparable damage to the enterprise. Based on the disk S.M.A.R.T data, the machine learning method is used to predict the disk failure, and better prediction results are available. However, due to the limitation of the S.M.A.R.T ability, only the information contained in a single time point leads to poor prediction, which cannot reach the industrial application level.

In this paper, the timing information is introduced into the disk prediction model, including data timing feature processing and timing model optimization. At first, a data timing allocation strategy is proposed to dynamically allocate time series data. A time series feature processing algorithm is proposed to extend the S.M.A.R.T features. In the prediction part, a timing weight random forest is proposed. Based on the time series characteristics of the sample data, different weights are initialized for different decision trees in the random forest. The mean and variance are used to fit the time series data. A negative feedback update model is proposed. A posteriori decision tree is introduced in the random forest; the negative feedback information is added based on the historical prediction data, and the weight update algorithm is proposed. Based on the above, the disk failure prediction prototype system was designed and developed, which is now deployed in large-scale data centers.

Through analysis and experiments, the time series prediction model proposed in this paper is superior to the traditional model. Compared with the traditional model, the recall rate increased by 11.13% under the same false alarm rate, and the false alarm rate was reduced by 52.0% under the same recall rate. Applying prediction to the disk scrub, the mean time to detection is decreased by 152.6%, when the accelerated status time is 5.0%, and the mean time to detection is decreased by 217.3% when the disk scrub load increases by 4.8%. It can greatly reduce the disk scrub overhead and reduce the scrub cost.

**Key words:** Disk failure prediction, S.M.A.R.T technology, disk scrub, decision tree

## 目 录

摘 要.....	I
Abstract.....	II
1 绪论	
1.1 研究背景.....	(1)
1.2 国内外研究现状.....	(2)
1.3 本文的主要工作.....	(4)
1.4 本文组织结构.....	(5)
2 相关背景	
2.1 磁盘 S.M.A.R.T 技术.....	(6)
2.2 决策树算法.....	(9)
2.3 特征选择算法.....	(14)
2.4 磁盘巡检技术.....	(15)
2.5 本章小结.....	(17)
3 时序随机森林模型	
3.1 时序数据处理.....	(18)
3.2 时序模型优化.....	(23)
3.3 本章小结.....	(28)
4 磁盘故障预测原型系统的设计与实现	
4.1 原型系统概述.....	(29)
4.2 磁盘数据采集系统.....	(32)
4.3 故障预测系统.....	(34)
4.4 磁盘巡检系统.....	(36)

# 华中科技大学硕士学位论文

---

4.5 中心监控系统.....	(40)
4.6 本章小结.....	(45)
5 系统分析测试	
5.1 实验环境.....	(46)
5.2 数据集与特征选择.....	(46)
5.3 功能测试.....	(49)
5.4 评价指标.....	(52)
5.5 性能测试.....	(56)
5.6 本章小结.....	(61)
6 总结与展望	
6.1 本文工作总结.....	(63)
6.2 未来研究工作展望.....	(64)
致 谢.....	(66)
参考文献.....	(67)
附录 1 攻读学位期间参与的科研项目.....	(74)
附录 2 攻读学位期间取得的科研成果.....	(75)

## 1 绪论

### 1.1 研究背景

近年来,随着互联网新兴技术爆炸式发展,产生的数据日益增长,对数据存储的要求越来越高,各企业的数据中心也更加庞大<sup>[1,2]</sup>,其中硬盘仍为主流存储设备。

数据中心中硬盘故障往往造成严重的影响,一般导致应用服务不可读写,从而给用户或者是服务提供商带来巨大损失<sup>[3,4]</sup>。在大型存储系统中,包括硬盘、CPU、网卡等硬件故障是常见故障类型。虽然存储系统中单块磁盘的使用年限较高,但在海量数据中心中,硬盘的故障频率往往是最高的、经常需要进行数据备份和替换<sup>[5-8]</sup>。

在大规模数据中心中,由于磁盘硬件的数量很大,其损坏率往往是所有硬件之首。根据微软统计关于其数据中心的数据显示,在各数据中心中硬盘故障占了所有硬件损坏的 78%<sup>[9]</sup>。关于磁盘故障,影响磁盘可靠性的因素非常多,而且磁盘扇区级别的故障也非常频繁,如磁盘温度,湿度,负载高低,运行时间,潜在扇区错误故障都将可能导致磁盘异常,造成记录的用户数据丢失<sup>[10]</sup>。

为提高存储系统可靠性,海量存储系统普遍趋向于采用被动预防的机制,如副本或纠删码的方法。如谷歌文件系统Google File System (GFS)<sup>[4]</sup>与开源的分布式存储系统Hadoop Distributed File System (HDFS)<sup>[11]</sup>均采用数据三副本存储策略,通过冗余来提供高可靠性。另外,纠删码<sup>[12]</sup>技术通过对原始数据通过校验算法计算校验数据并额外保存,在原始数据部分丢失时根据额外保存的校验重构原始数据,相对多副本冗余存储机制而言,纠删码的提出可使得存储空间利用率更高,与此同时其数据恢复时的重构开销往往令人望而却步<sup>[13,14]</sup>。

近些年,人们提出了利用硬盘的 S.M.A.R.T 技术<sup>[15]</sup>来主动预测硬盘故障。S.M.A.R.T 技术的全称为“Self-Monitoring, Analysis and Reporting Technology”。具体为磁盘的“自我监测、分析和报告技术”,它将监控硬盘的重要的健康属性指标并记录,同时为每个指标设定健康阈值;如果有指标低于阈值则认为硬盘可能发生故障并发出预警信息给操作系统或用户,系统收到预警信息后可提前迁移或备份该硬盘的数据<sup>[16]</sup>。目前,几乎所有的硬盘都支持 S.M.A.R.T 技术,但 S.M.A.R.T 技术采用基于阈值的简单算法,没有考虑 S.M.A.R.T 各种属性之间的关联关系,并且它



是基于属性实时数据进行预警，没有考虑到海量历史数据的特征分析，因此告警效果非常有限。但是磁盘的 S.M.A.R.T 记录反应了硬盘的健康状况，以及磁盘的一些性能指标，这些数据对于预测硬盘故障仍然是非常有意义的<sup>[15]</sup>。

随后基于 S.M.A.R.T 属性值，利用统计和机器学习方法对磁盘故障进行预测的技术不断兴起<sup>[17-19]</sup>。但受限于数据量和对磁盘故障模式，以及磁盘型号复杂过多的局限，预测效果都不佳。本文通过决策树模型，引入时间变化率特征和场景特征，结合磁盘巡检应用场景，在大规模云存储环境中，取得了工业适用的预测效果。

综上所述，存储介质作为云环境各项业务的基石，是最需要保证可靠性的部分。本文重点着重于建设大规模的磁盘故障预测系统，利用多种模型、算法来进行提前故障预测，与实际的磁盘巡检技术相结合。在大规模数据中心中部署故障预测系统，在长期的统计分析实验数据后，证明系统可提高存储系统的可靠性。

## 1.2 国内外研究现状

1992 年，IBM 最早提出通过对磁盘运行状态的监控，来评估磁盘的安全状况，并且将评估结果报告到操作系统以及用户。后来该技术被写入 ATA-3 行业标准，正式命名为 S.M.A.R.T 技术，即“自我监测、分析及报告技术”。S.M.A.R.T 实时采集磁盘的参数信息，并且与厂商设计的磁盘危险参数阈值进行比较，从而来判断磁盘是否处于安全状态。但是各厂商设定的磁盘各参数阈值不尽相同，厂商设置的阈值参数在实际中参考意义比较有限，许多研究者基于 S.M.A.R.T 数据采用机器学习的方法，进行磁盘故障预测方向的研究<sup>[20]</sup>。

Bairavasundaram 等<sup>[21, 22]</sup>研究发现随着磁盘的厂商、型号、接口、容量等属性不同，磁盘和其适配性校验与不匹配的概率与潜在磁盘扇区错误的比例有显著关系。潜在所有的磁盘型号中，潜在扇区错误随着磁盘容量的增长呈现正比例递增的趋势。

谷歌公司的 pinheiro 等<sup>[23]</sup>使用传统的基于统计方法对磁盘故障进行分析，采集超 10 万块磁盘时间跨度约 300 天的磁盘 S.M.A.R.T 数据信息，他们发现磁盘的故障率与磁盘的温度与磁盘的利用率都几乎没有关联。同时发现磁盘状态监测属性中的其他属性值与磁盘故障率有较大的关联，如扇区扫描错误属性（scan errors），磁盘扇区重映射计数属性（reallocation counts），磁盘离线扇区重新分配计数属性（offline

reallocation counts），磁盘试用计数属性（probational counts）。研究发现发生过第一个扫描错误的磁盘，比没有扫描错误的磁盘在接下来的 60 天内发生故障的概率要高出 39 倍，可见扇区错误存在着时间的相关性。其他的三个 S.M.A.R.T 参数分别为高出原来的 14 倍，和 21 倍以及 16 倍。由于厂商设置的磁盘阈值差异等原因，大部分的故障磁盘没有 S.M.A.R.T 预警发生，因此他们认为不能根据磁盘厂家设置的阈值，来建立准确的磁盘故障模型，S.M.A.R.T 模型更适合用预测磁盘的整体集群趋势。

为了提升基于 S.M.A.R.T 的预测的精度，Hamerly 和 Elkan<sup>[24]</sup>运用两种贝叶斯方法在昆腾厂商的磁盘进行试验。数据集包括 1934 块好的磁盘与 9 块在 48 小时内将要故障的磁盘，共包括带时间标记的 94022 个好盘标签数据与 34 个坏盘标签数据的 S.M.A.R.T 属性值记录。他们发现有监督的贝叶斯分类器对不相关的属性具有很好的鲁棒性。在他们的研究结果中，NBEM 算法在低维度上的效果要好于高维度的效果，只有 3 个 S.M.A.R.T 属性项的数据集，实现可以获得 33% 的检测率和 0.36% 的误报率。标准朴素贝叶斯分类器在高维情况下表现更好，利用 9 个 S.M.A.R.T 属性项的预测效果最优，可以得到 56% 的检测率和 0.82% 的误报率。

Murray<sup>[25-27]</sup>等用多种方法进行了磁盘故障预测的实验，其中包括支持向量机、非监督聚类、两种非参数统计测试，同时基于多实例学习框架改进了一个新算法，名为多实例朴素贝叶斯。对朴素贝叶斯的特殊改进是为了满足低的误报率（FAR）。基于中同样的数据集，S.M.A.R.T 数据在中被当作时间序列数据，他们运用隐马尔可夫模型和半马尔可夫模型来提升预测的效果。试验数据集包括 369 块磁盘，其中好盘为 191 块。数据集中的好盘都在制造商监控的环境下通过了验证测试，数据集中的故障盘为用户报修返厂后确认的磁盘。最终，磁盘实用 mi-NB 方法可以获得 50% 的检测率和 0% 的误报率。

张超<sup>[28]</sup>提出了 T2us 算法尝试进行磁盘故障预测分析并且取得较为理想的效果。T2us 算法主要包括两个阶段：第一步是磁盘运行环境的判断，然后是磁盘状态的相容性判定。在第一阶段中，对故障磁盘的环境进行建模，选定故障磁盘的环境变量，在此基础上进行辅助判断。在第二阶段中，基于磁盘 S.M.A.R.T 技术中的多种数据，潜在故障扇区数，重定向扇区数作为集合，进行拓展的自学习算法进行判定。实验结果表示算法故障检测率为 45%-52%。

王刚<sup>[29]</sup>团队与百度公司合作，提出使用分类和回归树进行预测磁盘故障，将决策树算法用作主要的预测算法。他们的实验中自行采集的 51703 磁盘共 2 个月的数据，其中每块磁盘包括 24 项 S.M.A.R.T 原有属性，与其他 8 项型号等属性，通过原有的 S.M.A.R.T 属性进行自定义训练。在决策树模型中进行训练后，他们的最优结果为误报率 1%，误报率为 93%。

Farzaneh<sup>[30]</sup>使用主动预测磁盘故障的方法，尝试提高存储系统的可靠性。磁盘巡检速率的提高可缩小巡检周期，降低坏块在系统中的停留时间，从而提升存储系统的可靠性，但是同样也会对磁盘业务访问造成一定的影响。作者通过对磁盘进行故障预测，然后对可能会发生的坏盘进行磁盘巡检速率的提高，从而降低了对前台业务访问的代价，同时也能及时发现潜在故障扇区。根据上述研究，总结关于磁盘故障预测各种方法、数据以及性能总结在表 1-1：

表 1-1 现有磁盘故障预测研究总结

研究者	好盘	坏盘	算法	属性数	召回率	误报率
Bairavasundaram 2007	-		Satastic	-	-	
pinheiro 2007	超过 10 万		Satastic	-	-	
Hamerly& Elkan 2001	1927	9	NB	9	56.0%	0.82%
			NBEM	3	33.0%	0.36%
Murray et al.(2005)	178	191	NP Rank Sum	4	52.8%	0.70%
			Mi-NB	25	50.0%	1%
张超 2008	-		T2us	-	45.0%	-
Wang Gang 2017	98060	243	Decision tree	20	85.1%	0.70%
Farzaneh 2017	36368	433	Decision tree	11	70.0%	2%

## 1.3 本文的主要工作

本文主要在大规模磁盘故障预测中，结合磁盘时序数据，致力于提升预测准确率；并且结合实际磁盘巡检场景，从而提升存储系统的可靠性。本文将对预测模型和算法进行优化，在数据处理上利用时序数据，并且结合云存储系统中的实际应用，将磁盘故障预测与应用进行结合。

本文的主要工作如下：

- (1) 提出时序随机森林模型，并应用于磁盘故障预测系统
- (1) 优化决策树模型，利用数据时序性进行建模、模型更新
- (2) 结合磁盘巡检应用，减少巡检代价提升系统可靠性
- (4) 设计磁盘故障预测巡检原型系统，并应用于数据中心中

## 1.4 本文组织结构

本文共六个章节，各章节内容如下所示。

第一章节是绪论，介绍大规模磁盘故障预测的相关背景，以及国内外相关研究的最新成果，并简述本文的主要研究内容和文章框架。

第二章节是相关背景介绍，包含了磁盘 S.M.A.R.T 技术，决策树算法及其原理，特征选择算法实现方式，以及磁盘巡检有关技术共四个部分。

第三章节是本文核心创新点，内容为时序数据特征处理与预测模型构建，包含了本文提出的时序数据分配策略，与时序特征处理算法；在模型构建中，包含本文提出的时序决策树投票算法与负反馈更新策略。

第四章节是关于大规模磁盘故障系统的原型系统，包含数据中心的磁盘数据采集、算法模型、磁盘巡检和监控部署模块。为后来的研究者提供必要的帮助。

第五章节是关于本文系统的实验测试，从整体的角度进行梳理磁盘故障预测过程的梳理，包含磁盘故障预测和磁盘巡检两部分。最后有对实验的评价指标进行描述分析，证明算法模型确实有效。

第六章节是总结与展望，指出本文研究的不足之处，以及对未来工作的展望，和在磁盘故障预测相关课题提供研究方向。

## 2 相关背景

在进行大规模磁盘故障预测讨论前，需要对相关背景技术有所了解。本章将重点描述本文研究目标和创新点的相关背景知识，包括磁盘 S.M.A.R.T 技术背景和发展，随机森林算法和决策树算法原理，特征选择算法的计算，磁盘巡检技术以及应用等。

### 2.1 磁盘 S.M.A.R.T 技术

磁盘 S.M.A.R.T 技术是一种自动的磁盘状态预警和系统监测的规范<sup>[15]</sup>，它通过磁盘内部硬件监测指令对磁盘信息进行监控，如温度、磁头、马达、盘片、电路等数据，然后将采集到的状态信息与磁盘生产厂商的安全阈值相比较，来判断磁盘自身的健康状况。若采集到的状态信息已超出预设的安全范围，若磁盘判断自身异常，将会磁盘将相应的状态保存在 S.M.A.R.T 数据中。磁盘同时自身可能会尝试自动修复故障，并且将修复记录也保存在 S.M.A.R.T 数据中，提供给磁盘使用者进行处理。目前除了极早出厂的一些硬盘之外，现在大部分的磁盘都已经支持 S.M.A.R.T 技术进行磁盘监控。

#### 2.1.1 磁盘 S.M.A.R.T 技术简介

S.M.A.R.T 硬盘监控技术起源于 1992 年，原型为 IBM 为提升计算机 AS/400 的硬盘阵列 IBM9337 的存储可靠性，设计使用的故障分析探测技术。后台这项技术被命名为故障预警分析技术（Predictive Failure Analysis），它是通过在磁盘固件中测量几个重要的运行状态参数来评估磁盘的健康状况。在监测软件<sup>[31]</sup>中对磁盘持续监测后会产生两种可能的结果，“磁盘目前安全”或“磁盘将会发生故障”。后来，由计算机制造商 Compaq 和硬盘制造商 Seagate、Quantum 等一同提出了名为 IntelliSafe 的相关技术。由于 IntelliSafe 技术在设计、实现上的灵活性更好，Small Form Factor 委员会接受了该方案，并将此技术正式更名为 S.M.A.R.T. 技术，把这项技术进行标准化并推广至 ATA-3 行业标准<sup>[32]</sup>。

S.M.A.R.T 技术成为行业标准后，各磁盘生产厂商开始遵循以此项标准进行设计、



生产磁盘。该标准规定了磁盘厂家在生产磁盘时必须满足的一些条件：磁盘生产过程中需要加入相关的磁盘监控属性，设置规定的标准阈值；磁盘的 S.M.A.R.T 状态监测功能可以被用户随时开启和关闭；在各种操作系统平台下，都可以正常使用 S.M.A.R.T 技术，若该系统或者设备对磁盘的 S.M.A.R.T 技术不支持时，用户应该得到相应的提示信息；在磁盘正常工作时，用户可以读取磁盘各项正确有效的状态监测信息，从而来确认磁盘设备是否处于正常的工作状态，如果在运行过程中发生了异常或者故障，S.M.A.R.T 技术支持预警、警告来告知用户<sup>[33]</sup>；在磁盘支持 S.M.A.R.T 技术以及操作系统默认该技术开启时，磁盘出现异常或者故障时，磁盘应提醒用户进行磁盘修复或者尝试自修复以及进行数据备份<sup>[34-36]</sup>。

## 2.1.2 S.M.A.R.T 检测参数

下面将列出磁盘设计生产商一般都会使用的一些 S.M.A.R.T 参数，包括中英文名和含义如表 2-1 所示。不同的参数标识不同的状态监测数据，这些参数在协议中规定磁盘设计生产商必须遵守的规则，但有些状态监测参数在不同的硬盘制造商中用不完全相同的定义和计量方法，磁盘制造商有时也会添加一些专有的监测参数进行磁盘监控。这些 S.M.A.R.T 监测参数较多，本节不再一一列出。

表 2-1 S.M.A.R.T 的原始检测属性和含义

属性 ID	十六进制	中英文名	属性说明
1	0x01	Read Error Rate 底层数据读取错误率	硬盘表面读取数据时发生的错误率
3	0x03	Spin-Up Time 盘片启动时间	盘片加速到稳定正常运行速度时间
4	0x04	Start/Stop Count 电机起停次计数	一个盘片启动关闭周期的统计值
5	0x05	Reallocated Sector Count 重定向扇区计数	记录损坏而被重映射的扇区计数
7	0x07	Seek Error Rate 寻道错误率统计	磁头发生寻道错误的事件概率

## 华中科技大学硕士学位论文

9	0x09	Power-On Hours 硬盘通电时间	硬盘自运行来的累计通电时间
12	0x0c	Power Cycle Count 设备开关计数	该属性表示硬盘电源开关循环计数
13	0x0d	Soft Read Error Rate 软件读取错误率	操作系统读取数据时的出错率
190	0xbe	Airflow Temperature 气流温度	计量硬盘内气流温度
191	0xbf	G-sense Error Rate 加速度错误率	计量对硬盘做成损害的冲击次数
194	0xc2	Temperature 温度	计量硬盘的温度
200	0xc8	Write Error Rate 写入错误率	写入一个区域时发现的错误的计数
204	0xcc	Soft ECC Correction 逻辑 ECC 纠正	记录由软件 ECC 更正的错误计数
240	0xf0	Head Flying Hours 磁头飞行时间	磁头处于定位中的时间
254	0xfe	Free Fall Protection 自由跌落保护	对“自由落体事件”检测计数

S.M.A.R.T 包含 6 项属性，属性号、属性值、历史最差值、阈值、原始值、状态标识。它实时收集磁盘运行时的健康状态数据，并基于收集的数据对磁盘故障进行预警。出于降低误报带来的高成本的目的，厂商会设置相对较大的阈值。在实际使用中基于阈值的方法得到的预测效果并不理想，故障检测率（FDR）在较低的误报率（FAR=0.1%）时只有 3-10%。

正是由于基于阈值方法的故障预测效果不理想，统计和机器学习方法被引入到磁盘故障预警中<sup>[37-39]</sup>。综合 S.M.A.R.T 属性的表征能力，与 S.M.A.R.T 属性联系挖掘潜藏的故障模式，使得统计和机器学习用于磁盘故障预警变得更加具有实际意义。

## 2.2 决策树算法

故障检测在数据挖掘中是一项常见而十分重要的问题之一，包括深度学习、机器学习和统计学等领域的学者已提出许多种的解决方案。其中故障检测问题运用最广泛的几种方法包括决策树，深度网络，贝叶斯分类和 SVM 等。

决策树作为一种数据挖掘算法，被广泛的应用于客户分析、市场商业策略制定、医疗研究、银行个人信贷风险策略等方面<sup>[18, 19, 40]</sup>。在 2007 年评选的十大算法中，决策树算法占了两席，分别是 C4.5 和 CART。相比于其他算法，决策树有着易使用、快速、易于理解等特点，尤其是当特征变量为离散变量时，决策树有着非常优异的表现。

使用决策树进行故障检测有许多优点，首先其可解释性强、构建简单、结构清晰，类似于人类的思维脉络。其次，决策树计算方法时间复杂度很低，对故障探测等二分类问题处理效率很高。最后，决策树方法的分类准确率很高，通过剪枝等技术手段可应对各种异常噪声数据，使得系统有良好的鲁棒性。故障检测领域中，决策树在深度学习网络等机器学习算法中脱颖而出，得到学者广泛的使用。

### 2.2.1 决策树原理

决策树方法最早产生于上世纪 60 年代，由 Hunt 等人于 1966 年时提出的概念学习系统<sup>[41, 42]</sup>不断发展而来。在后续发展中对后来有更大影响的是 Ross Quinlan 与 1986 年提出的另一种改进方法，名为 ID3 算法<sup>[43]</sup>的优化决策树模型。它的主要变化为使用属性信息增益对决策树的节点进行选择<sup>[44, 45]</sup>，此算法主要优势是减少决策树的纵向深度，但忽略了决策树的横向膨胀。在此基础上，后来很多学者提出对 ID3 算法的改进，其中 Ross Quinlan 后续提出的 C4.5 算法影响最为深远<sup>[46]</sup>。经典的决策树分类算法除了 ID3 算法与 C4.5 算法之外，还包括 CART 算法等，下面将进行介绍其原理和计算公式。

#### (1) ID3 算法

在构造决策树时，首要需要选取最佳分裂属性，其原则是被分裂属性能够有效地反映原有数据集合的特征。ID3 算法是构造决策树最经典的算法之一，它在每次节点分裂时计算节点分裂前后的信息增益，将样本集合的信息增益用于分裂的判断标



准，从而完成数据样本的划分<sup>[47, 48]</sup>。在使用 ID3 算法计算信息增益前，需要了解先了解信息论中最基本的概念信息熵。信息熵一般作系统有序或者混乱程度的度量指标，其计算方法如式(2-1)与式(2-2)：

$$Info(D) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i) \quad (2-1)$$

$$Info(D) = - \sum_{i=1}^n p(x_i) \log_2 p(x_i) \quad (2-2)$$

其中  $n$  为特征个数， $p(x_i)$  为  $n$  个特征中属于  $x_i$  的样本在总样本中的比例，记总训练集样本为  $D$ ，那么类别为  $x_i$  的样本数为  $d_i$ ，则有式(2-3)：

$$p(x_i) = \frac{|d_i|}{|D|} \quad (2-3)$$

信息熵  $Info(D)$  表示将原有数据集  $C$  不同的类一一区分来所需的信息量，它实际上就是信息论中的熵，熵表示为不确定性的度量值，如果某个数据集的类别不确定性程度越高，那么熵的值就越大。

在 ID3 算法分裂训练集选择最佳属性时，需要遍历每一个属性，来获取最大的信息增益。其中对某一个属性  $R$  进行分裂时，需要求出对应的信息增益。信息增益与对属性  $R$  分裂的值有关，因此在分裂时必须将对应数据的所有值进行离散化。假设数据集  $D$  的属性  $R$  的值分成  $K$  组  $\{D_1, D_2, D_3, \dots, D_k\}$ 。在对属性  $R$  分裂后，将数据集  $D$  不同的类区分开，需要的信息量如式(2-4)：

$$Info_R(D) = \sum_{j=1}^k p(x_j) * Info(D_j) \quad (2-4)$$

其中  $k$  是属性  $R$  值的所有取值可能个数， $p(x_j)$  记为对应样本占有所有属性  $R$  的比例， $Info(D_j)$  记为每一个分裂后的样本集合  $D_j$  的信息熵。

信息增益的定义为，对属性  $R$  分裂前后的信息量之差，如式(2-5)：

$$Gain(R) = Info(D) - Info_R(D) \quad (2-5)$$

信息增益  $Gain(R)$  表示属性  $R$  分类时带来的信息量，每次分裂希望得到最大的属性增益，需要对可分裂的属性计算  $Gain(R)$ 。这样我们每次分裂都是尽量的纯，最可能将不同的类区分来。因为对于所有的  $Info(D)$  是相同的，所以我们将求最大的

$Gain(R)$ 问题等同于求最小的 $Info_R(D)$ 。

同时 ID3 自身特点也存在一些局限性，如针对属性集中的连续型属性，必须经过离散化之后才能被进行分裂计算；节点划分过程中可能会由于子集规模过小，导致统计特征不充分等，后续的研究学者也提出了许多优化改善算法<sup>[49-51]</sup>。

## (2) C4.5 算法

C4.5 算法是 Ross Quinlan 本人对 ID3 算法的优化版本<sup>[46, 52]</sup>，它最大的不同在于节点每次分裂时，采用的标准为信息增益率而非信息增益，进行特征的划分。C4.5 算法还使用了后剪枝技术进行决策树的优化，从而减少树纵向过深或者横向过粗的问题，算法可以处理样本数据的连续性特征，在部分数据样本特征缺失时也可进行适当处理，近期针对 C4.5 也有一些研究型成果出现<sup>[53, 54]</sup>。

C4.5 算法计算信息增益率来进行最佳属性的划分，其中分裂信息计算如式(2-6)，信息增益率的计算如式(2-7)：

$$SplitInfo_R(D) = - \sum_{j=1}^k \frac{|D_j|}{|D|} \times \log_2 \left( \frac{|D_j|}{|D|} \right) \quad (2-6)$$

$$GainRatio(R) = \frac{Gain(R)}{SplitInfo_R(D)} \quad (2-7)$$

C4.5 算法在使用时，节点每次分裂计算信息增益率  $GainRatio(R)$ ，将待分裂集合中最大的那一个属性进行分裂。如果一个属性的取值很多，那么  $SplitInfo_R(D)$  会更大，从而使得信息增益率  $GainRatio(R)$  变小，这样可以解决 ID3 算法分裂时容易选择取值较多属性的弊端。

## (3) CART 算法

CART 算法由 Breiman 等人提出<sup>[55, 56]</sup>，在统计学领域得到广泛地运用<sup>[43]</sup>。它在节点分裂时主要使用二分递归的思路进行分割，最终输出的结果是一颗结构简单没有多余分支的二叉树。CART 算法在分裂时都会将数据样本划分为两个子集合，分别用两条分支来代表每一个非叶子集合的分裂情况，而且样本数据的分支节点每次裂时仅有“是”或者“否”两种结果，在样本数据集合中有多种特征时，在数据预处理部分时还需将所有的属性值区分为两部分。

CART 算法在进行样本属性分裂时采用基尼 ( $Gini$ ) 系数作为判断依据， $Gini$  系数的计算如下式：

$$Gini_R(D) = 1 - \sum_{j=1}^k p_j^2 \quad (2-8)$$

其中  $k$  标识数据集  $D$  中属性  $R$  各类别的数目,  $p_j$  表示属性  $R$  中样本属于  $D_j$  的概率。每次在对属性区分的时候, 希望最大化  $Gini_R(D)$  从而得到最纯的分类。

### 2.2.2 随机森林

随机森林起始于 Leo Breiman 和 Adele Cutler 共同研究的一种集成方法<sup>[57-59]</sup>, 同时将该项技术注册为商标<sup>[60, 61]</sup>。随机森林通过组合多个弱分类器, 对不同分类器使用加权投票的方式, 从而整体模型有较高的准确率, 使得适合解决分类与回归问题。随机森林是机器学习领域重要的组成部分, 在工程实际应用中有广泛地使用。

随机森林由多颗决策树组成, 通过对训练集进行抽样为训练子集, 输入到每颗决策树中。分类时每颗决策树分别对训练样本的结果投票, 将不同的决策树预测结果加权, 从而形成整个随机森林的预测结果, 最终完成分类问题。由于随机森林具备分类精度高, 不容易过拟合鲁棒性较好, 目前在故障预测, 金融分类等领域具有广泛地运用。本章针对随机森林的构建和优化作出分析, 使其适应磁盘故障任务场景

由于上小节提到决策树的种种缺点, 不适合与大规模的分类问题, 于是将多颗决策树分类器组合成多分类器的思想就逐渐形成。在随机森林的构建中, 对森林中每个决策树单元都赋予初始权重, 以投票的方式输出分类结果, 其主要的过程有三步, 首先是对训练样本子集进行抽样, 其次是生成每颗决策树, 最后是按照不同的决策树的预测结果, 加权计算成为随机森林的输出。

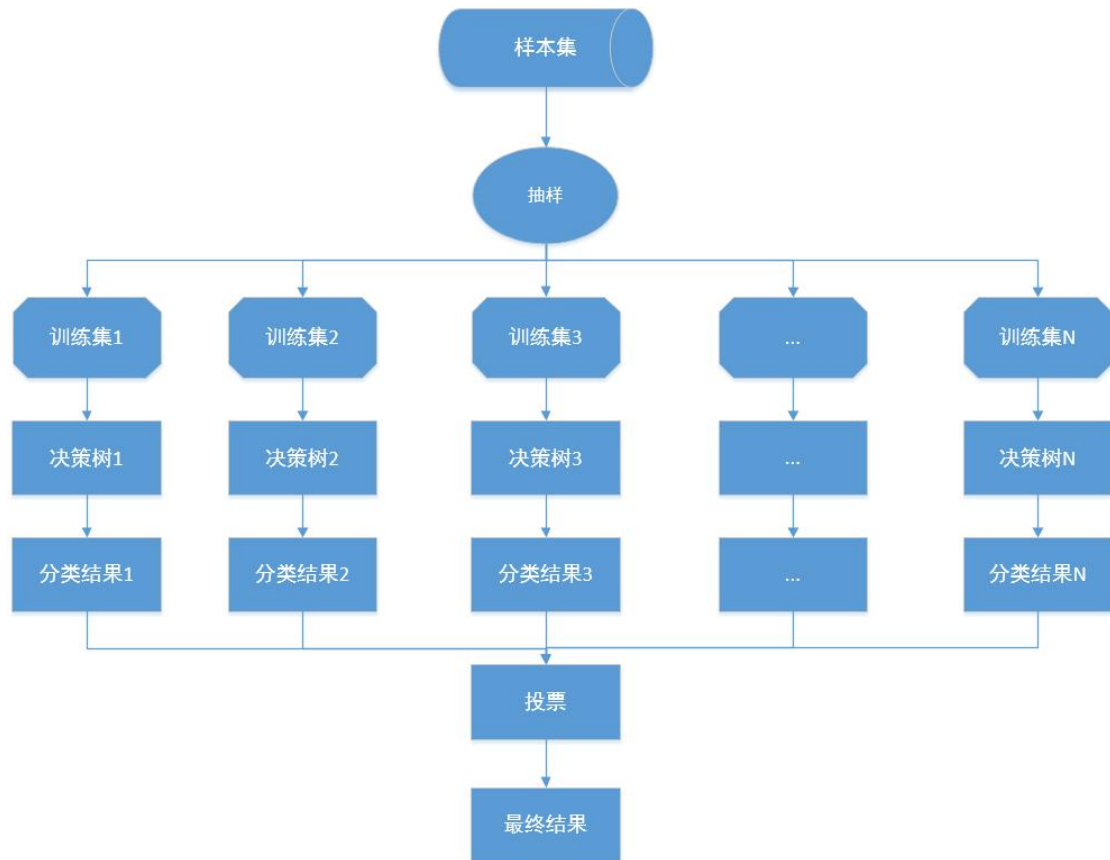


图 2-1 随机森林逻辑架构图

随机森林主要使用 Bagging 有放回地随机抽样，对随机森林的  $N$  颗树进行  $N$  次随机抽样形成对应个数的训练子集，其中根据设定的不同每次的训练数据大约是样本训练集的  $2/3$ 。其逻辑架构如图 2-1 所示，一个完整的随机森林的生成流程如下：

（1）在预处理时将原始训练集进行  $N$  次数据采样，在每次采样时有放回地随机选出  $M$  个样本，最终共生成  $N$  组训练集；同时对应随机森林中的每棵决策树，使用对应的  $N$  个训练集训练；

（2）对于随机森林中的决策树，每次分裂时按照设定的算法进行分支节点的分裂，按算法不同分别有信息增益、信息增益比、基尼指数等方法；

（3）每棵树从树根进行判断叶子节点或分支节点，对分支节点进行同样分裂步骤，直到该节点的所有训练样例都属于同一类，返回该决策树；

（4）将上述每次生成的决策树共同组成随机森林网络。综合其中每一颗决策树的预测结果，加权计算成为随机森林的输出。

## 2.3 特征选择算法

特征选择的目的是寻求分类算法的有效特征信息，避免样本数据特征过多，大量无关的数据参与计算导致耗时过长。在模型训练中，由于特征数目过多而其中部分属性可能表现出相关性，使得特征效用重叠。特征子集的优良由多种特征评估算法进行判断<sup>[62]</sup>。

主元分析法 PCA(Principal Components Analysis) 是将高维特征降低到低维特征的一种有效方法，在多元统计分析方法中占有举足轻重的地位<sup>[63]</sup>。它将原本是高维的大型样本数据集映射到低维空间进行求解，从而去除原有数据中的无关冗余信息，可以大幅度降低计算开销。PCA 作为一项基本的数据处理技术，应用于机器学习中多个领域，如计算机视觉领域进行图像属性域处理，人脸识别中对无关特征排除，检测领域中重要信息提取等<sup>[64]</sup>。

PCA 算法是将高维数据映射到低维数据，在保证模型精度与数据表达性的情况下，减少数据的维度从而降低模型计算的复杂度。算法的主要思想是将原有数据从较大的  $n$  维特征映射成为较小的  $k$  维数据，得到的  $k$  维数据是通过计算而重新构造出来的正交特征组集合。算法首先求出输入向量的特征值和特征向量，然后选择出其中的最大  $k$  个特征值，通过其对应的  $k$  个特征向量而得到最终的输出向量  $Y$ ，假设 PCA 算法的输入  $n$  维数据集  $X = \{x_1, x_2, x_3, \dots, x_n\}$ ，输出  $k$  维数据集  $Y = \{y_1, y_2, y_3, \dots, y_k\}$ 。其中  $x_i$  为样本集中第  $i$  维的属性集合，首先是将  $x_2$  去中心化，使各项属性减去其均值  $\bar{x}$ ， $\bar{x}$  的计算如式(2-9)：

$$\bar{x}_i = \frac{1}{N} \sum_{j=1}^N x_{i,j} \quad (2-9)$$

其中  $x_{i,j}$  是第  $j$  条数据的第  $i$  个属性， $N$  为样本的总条数；在计算出各属性的均值后，将原数据集  $X$  去中心化，步骤如式(2-10)：

$$x_i = x_i - \bar{x}_i \quad (2-10)$$

然后计算协方差矩阵  $C$ ， $C$  的计算如式(2-11)：

$$C = \frac{1}{N} X X^T \quad (2-11)$$

求出协方差矩阵后，通过该矩阵求得特征值  $\lambda$  和特征向量  $\gamma$ ，如式(2-12)：

$$C * \gamma = \lambda * \gamma \quad (2-12)$$

之后对特征值进行排序，按规则在所有的特征值中选择最大的  $k$  个特征值，计算出对应特征向量集合，再得到对应的特征向量矩阵  $P$ ；最后的  $k$  维数据  $Y$  表示如式(2-13)：

$$Y = P * C \quad (2-13)$$

通过计算数据矩阵的协方差矩阵，这样就可以将数据矩阵转换到新的空间当中，实现数据特征的降维。

## 2.4 磁盘巡检技术

### 2.4.1 磁盘巡检技术简介

磁盘巡检是用于提高数据可靠性的工具，可以通过周期性的扫描来完成对磁盘中潜在故障扇区的发现<sup>[65, 66]</sup>。

潜在故障扇区是一种常见的磁盘故障，在磁盘某扇区发生故障时，当磁盘访问该扇区时会导致磁盘变得不可读或不可写，而其他的磁盘扇区不会有异常，工业上处理磁盘潜在扇区故障，一般采取用巡检的方法，最基础的巡检方式是依次读写磁盘中每一个的扇区，磁盘厂商也开发了专用的巡检命令优化磁盘的巡检方式<sup>[67, 68]</sup>。

在磁盘巡检过程中，如果该扇区不是故障扇区则巡检通过；如果该扇区数据出现错误，说明此扇区存在异常，磁盘将会启动自修复程序。磁盘扇区自修复程序因磁盘生产厂家的设置有所不同，但一般会将不可修复的扇区进行重定向到备份扇区，从而维持磁盘的基本读写功能。希捷厂商设置在巡检发现坏块后，尝试进行数据重新写入，在重新写入仍然发现错误后，会将该扇区重定向到备份扇区。在故障扇区被重定向修复后，在下次访问到该扇区时，磁盘将会把对应备份扇区的数据返回。

巡检指令在 SCSI 指令集中为 VERIFY<sup>[69]</sup>，根据磁盘扇区地址长度的不同分为 VERIFY12 和 VERIFY16，其指令编号分别为 0xAFh 和 0x8Fh，指令结构如图 2-2 与图 2-3 所示。



Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (AFh)							
1	VRPROTECT			DPO	Reserved		BYTCHK	Obsolete
2	(MSB) LOGICAL BLOCK ADDRESS							
5	(LSB)							
6	(MSB) VERIFICATION LENGTH							
9	(LSB)							
10	RESTRICT ED FOR MMC-4	Reserved		GROUP NUMBER				
11	CONTROL							

图 2-2 Verify12 指令结构图

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (8Fh)							
1	VRPROTECT			DPO	Reserved		BYTCHK	Reserved
2	(MSB) LOGICAL BLOCK ADDRESS							
9	(LSB)							
10	(MSB) VERIFICATION LENGTH							
13	(LSB)							
14	Restricted for MMC-4	Reserved		GROUP NUMBER				
15	CONTROL							

图 2-3 Verify16 指令结构图

发起者通过向设备发送磁盘巡检指令，控制设备进行逻辑块的检查。每一个逻辑块的检查，包含用户数据和其他的保护信息两部分，基于发起者对 VRPROTECT 字段的填写格式。其中 LBA 为待巡检的目标扇区起始地址，LENGTH 为待巡检的长度，我们可通过对设备下发 SCSI 命令时，修改指令结构中命令控制块的数据段部分，对磁盘下发不同的巡检长度。

## 2.4.2 磁盘巡检应用

在磁盘巡检时，对系统前台的 I/O 有一定的影响<sup>[30]</sup>，不同的巡检参数对前台 I/O 的影响有所不同。巡检一般希望对前台的影响不要过大，所以会设置相应的检查频

率和时间间隔，以及每次检查的扇区个数等。

在存储中心中，频繁的巡检会对线上产生过大的压力，但同时周期太长的巡检会导致坏块在系统中停留时间过长<sup>[70]</sup>。平均坏块发现时间 MTTD (Mean Time to Detection) 被用来评估坏块在系统中停留时间，MTTD 越大代表坏块在系统中的停留时间越长，系统的可靠性越低。存储系统的 MTTD 一般与磁盘的巡检周期有关，在磁盘巡检周期越短时，全部扇区被扫描一遍的时间越短，同时 MTTD 也越短。

在存储系统设置巡检参数时，需要考虑到 MTTD 不能过长，同时也希望磁盘巡检的速率不能过快。所以在考虑设置巡检参数时，需要考虑磁盘的 MTTD 来提升存储系统的可靠性，也要平衡巡检对前台 I/O 的影响的损失。

## 2.5 本章小结

本章对本文使用到的相关技术背景进行描述，包括磁盘故障预测的数据来源，模型优化以及落地应用等。磁盘 S.M.A.R.T 技术为磁盘故障预测提供数据基础，决策树和随机森林算法为本文选用的主要模型，特征选择算法降低计算优化模型特征，将故障预测技术应用于磁盘巡检，可大幅度降低磁盘巡检的开销和降低坏块的故障探测时间。



## 3 时序随机森林模型

在大规模磁盘故障预测问题中，其核心研究点在于数据处理和预测模型优化，本章将重点描述这两部分内容，包括数据时序分配策略和时序特征处理算法，随机森林决策树时序权重投票算法、负反馈更新模型、以及决策树权重更新算法等，详细描述了上述算法的过程。

### 3.1 时序数据处理

在机器学习问题中，数据质量往往决定预测结果的上限。数据质量最重要的组成部分就是数据特征处理和训练、预测样本的分配。

面对时序数据问题，希望尽可能得到数据之间的时序信息，并将他们符合逻辑地安排到模型中。磁盘故障预测是一个典型的时序数据处理问题，一般此类问题需要对时序信息充分挖掘，样本数据集的时序变化体现了数据的逻辑结构。在训练数据以及测试数据的分配上同样需要考虑时序数据，减少因为数据先后关系影响模型质量。

特征处理步骤希望模型尽可能多的得到信息，又不希望引入无关信息，其中较为重要的是属性拓展和属性压缩。特征拓展要求不能引入与数据集无关的信息，以及重复的信息，一般通过数据集与数据集的关系相关而获得。而属性压缩可以通过PCA等特征提取的方式进行处理。

本文第二章已介绍了预测算法特征处理的基础算法PCA，本小结将介绍本文预测模型中采用的时序数据分配策略与时序特征处理算法，以及如何应用到实际模型的磁盘故障预测模型中。

#### 3.1.1 数据时序分配策略

在数据采集完成进行训练和测试时，需要按照时序预测问题的数据处理方法，对数据进行重新组合。磁盘故障预测为时序性分类问题，磁盘数据产生的先后顺序改变后，模型对同一块磁盘的分类结果随之会发生改变。

在磁盘故障预测问题中，磁盘样本不是在某一瞬间情况突然恶化的，往往是磁

盘表现出各种参数不稳定，在持续一段时间后磁盘状况才突然恶化，变为不可用的磁盘。而在磁盘表现为不稳定的一段时间中，如果能时刻关注磁盘的各种属性变化情况，那么很大程度就能得到坏盘模式信息。

在本文的磁盘故障预测问题中，将充分考虑磁盘数据的时序性。在数据分配时，同样使用数据的时序性方法，对不同阶段产生的数据对应模型处理的不同阶段。在具体模型训练时，首先将按照数据产生逻辑进行排列，按照时间先后顺序分别对应对于训练、验证以及测试数据集，从而符合数据流产生的时序性。其逻辑结构如图 3-1：

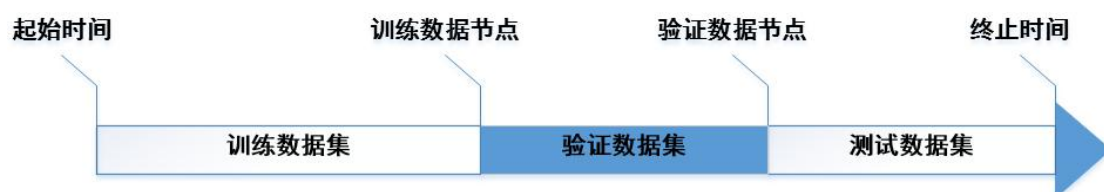


图 3-1 数据时序逻辑结构

如图 3-1 所示，我们将样本数据从起始时间到终止时间按照时间先后进行排列，按照比例分配最初的一段为训练数据，中间为验证集，最后为测试集。使数据的产生时间消费时间与验证时间，与数据进入模型的时间序列符合逻辑，模型能够理解数据的时序性。

不同的时间周期产生的数据其数据代表意义不同，在本次实验中我们将模型数据的流动与数据产生的周期同步，使得模型的训练在前，验证与测试在后。这样的设计会使得在磁盘样本特征发生变化时，时序模型能够发现它的变化情况，从而学习故障磁盘模式。

### 3.1.2 时序特征筛选算法

在实际模型预测中，数据的质量直接决定着模型的预测结果。在大规模磁盘故障预测问题中，由于原有样本是磁盘 S.M.A.R.T 属性，存在属性数目多，相关性大，表达能力不强等特点，需要对样本属性进行特征处理后才可进行训练预测。所以本小节设计将对原有样本数据进行改进，扩展部分属性的同时删去部分无用、重复的属性，从而增强模型数据的质量，提升预测的效果。

在磁盘故障预测问题中，原始 S.M.A.R.T 属性表达能力有限，无法表达属性变

化情况，但在对原有属性进行拓展时，需考虑数据的特点以及模型的特征，不能对属性添加无关的信息。如 S.M.A.R.T 5 号属性，它的变化率标识坏块数目在一定时间内的增加量，如果某一段时间增加量很多，说明磁盘的状况可能区域损坏。所以部分属性的变化率可以增加属性的表达含义，丰富原有样本数据。但是如 S.M.A.R.T 9 号属性磁盘的加电时间，传统意义的理解认为磁盘加电时间的绝对值与磁盘的损坏有关，一般该属性是常量累加的，所以该项属性的变化量始终为常数，所以该项数据的变化率意义不大。为在增加各属性的变化率后去除无用属性，需要在扩展属性后对属性进行部分特征处理。

所以本文的扩展属性将对原有属性集合添加变化率属性，对每个原有属性扩充增加一项变化率，其值等于上周期数据与现周期数据的差值，增加的具体方法如下。

对于原有的磁盘属性  $X_i$  表示为样本集合中第  $i$  条属性，需要对其扩充变化值属性  $Y_i$ 。记  $X_i = \{x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,n}\}$  其中  $n$  为属性的维数，同样地  $Y_i = \{y_{i,1}, y_{i,2}, y_{i,3}, \dots, y_{i,2n}\}$ ，其中  $2n$  为  $Y$  的属性个数；它们的转换如式 (3-1)：

$$Y_{ij} = \begin{cases} X_{ij} & j \leq n \\ X_{ij} - X_{i-1,j} & j > n \end{cases} \quad (3-1)$$

在属性扩充后，由于增加了太多的无关属性，需要对属性集合进行精简。冗余特征处理的方法在上一小节已经提过，此处将直接使用 PCA 进行特征处理。在 PCA 处理后剩余属性仍然冗余，本文将采用数据专家的建议，结合预实验手动提出一部分无关属性，从而保证模型数据的质量。对于磁盘故障预测模型特征工程时，采用下面的步骤进行处理，时序特征筛选算法逻辑架构如图 3-2 所示。

- (1) 统计所有磁盘共有的 20 个 S.M.A.R.T 属性；
- (2) 对 S.M.A.R.T 属性选取其 rawvalue 和 value 共 40 个作为特征；
- (3) 扩展基本特征，增加变化率特征共 80 个特征；
- (4) 选用 PCA 算法进行特征筛选出共 40 个特征；
- (5) 数据专家剔除多余属性，最终得到 19 个属性；

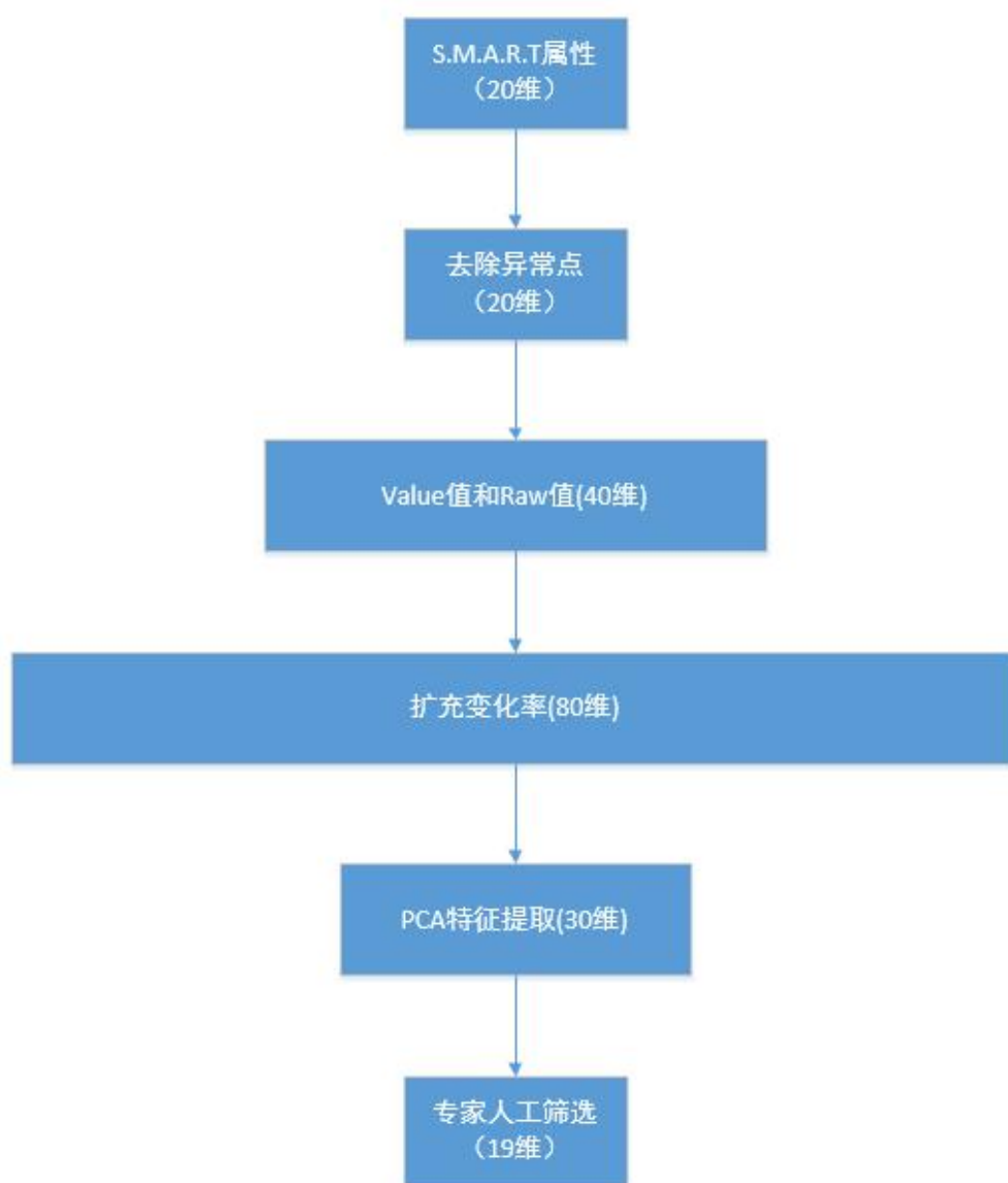


图 3-2 属性处理流程图

如图 3-2 为本系统进行特征处理的流程图，最初的 20 个 S.M.A.R.T 属性为数据集中共同拥有的 20 个属性。在第四步中，PCA 特征提取的方法为上小节的方法。在最后专家人工特征筛选中，主要采用两步进行无关特征筛选，第一步是手动剔除一些明显无关的属性，如硬盘加电时间属性的变化量等；第二步是在预实验中，通过测试实验筛选出无关属性。特征筛选后的 19 维属性如表 3-1 所示。

表 3-1 特征选择的 19 维属性

属性 id	英文名称	属性名称	被选择特征			
			value	raw	value_rate	raw_rate
1	Read Error Rate	读取错误率	√		√	
3	Spin-Up Time	盘片启动时间	√			
5	Reallocated Sector	重定位计数	√	√	√	√
7	Seek Error Rate	寻道错误率	√			
9	Power-On Hours	硬盘加电时间	√			
187	Uncorrectable Error	不可纠正错误	√		√	
189	High Fly Writes	磁头写入高度	√			
194	Temperature	温度	√			
195	ECC Recovered	ECC 恢复	√		√	
197	Pending Sector	待修复扇区数	√	√	√	√

如表 3-1 所示，为最终模型选择的 19 维属性，其中第一列是属性的标号，第二列是属性的英文名称，第三列是属性的中文名称，第四列是被选择是特征；其中第四列分为四小列，分别表示 S.M.A.R.T 属性 value 值、S.M.A.R.T 属性 raw 值、属性 value 值的变化值以及 S.M.A.R.T 属性 raw 值的变化值。其中变化值是通过人为计算，通过将同一块磁盘的前后两条数据的差值计算，然后重新设置为一条新的特征而形成。

在磁盘故障预测的建模阶段，我们使用时序数据分配策略进行数据分配，使用时序特征处理算法进行特征处理，两者共同作用保证模型对时序数据的抽象能力。在时序数据分配阶段，将样本数据按照时间分为训练集、测试集与验证集，从而保证模型各个阶段都可以得到数据的时序信息，可时刻感知数据的变化情况。在数据特征处理时，我们采用时序特征处理算法对所有样本集合的数据进行数据增加与修正。在特征处理后可增强 S.M.A.R.T 属性表达能力，反映各数据特征随时间变化情况，从而在数据特征层面丰富模型。

## 3.2 时序模型优化

在机器学习的分类问题中，分类算法模型的构造和更新是故障预测模型的核心。本小节着重于大规模数据故障预测模型的构建，以及对带有时间标签的时序类预测问题的处理，和模型更新问题的解决。在随机森林预测模型构建中，使用数据样本的时序性对每颗决策树初始化赋予不同的权重，使得单颗决策树接受不同时序的样本数据，同时其初始权重也将动态调整。时间样本越近的决策树其初始权重越高，反之则越远，随机森林将相对更关心时间较近的决策树的输出结果。在模型更新时加入负反馈决策树与权值更新算法，对预测错误的样本集合重新学习，对预测正确的决策树正向激励。在随机森林原有的各决策树中，增加一颗负反馈决策树用语训练历史预测失败数据，达到负反馈学习效果。同时模型更新时更新各决策树预测权重，按照历史预测结果对随机森林中各决策树权值进行动态调整。

大规模磁盘故障预测是时序预测问题，其数据样本带有时间标签且时间不同对模型的贡献不同。对于同样一块磁盘的数据特征，时间范围越近的数据其价值越高，对现在的磁盘情况影响越大。所以在对随机森林模型进行建模时需要充分考虑到数据的时序性，在模型的各个决策树中通过各决策树对应的权重反映。本文设计模型的时序特征中，对随机森林中对不同的决策树节点，其对应的不同样本数据设置不同的权重。对于磁盘特征越近的样本集合，其权重将会越高；磁盘特征越远的样本集合，其权重将会越低。在模型建立完成投票时，随机森林中各决策树的预测结果乘以该加权系数作为最终结果。

模型建立完成后根据时序数据样本进行故障预测，但随着数据量越来越多和时间不断累计，样本模式可能也会有所变化，预测模型同样需要时刻更新。在模型更新时需考虑上周期磁盘预测错误的情况，将其作为样本数据重新训练，在下一个时间周期将把该部分数据重新学习。为此本文设计了负反馈更新模型，将历史错误预测的样本集合重新生成样本数据集，并且在随机森林中新加入一颗负反馈决策树，在模型更新时对其权重进行调整。在模型更新时，通过对上周期的预测结果动态调整，如果上周期预测的准确率高，那么将其权重相应增大，反之则降低其预测权重。通过加入该负反馈机制，单颗决策树可动态学习错误预测数据，从而保证整体模型的准确率。



## 3.2.1 随机森林投票算法

在随机森林的构建中，最重要的部分是内部决策树的算法以及数据分配的流程。对应时序数据的预测中，其中与一般预测问题的处理最大的区别在于数据时间的连续性，以及对于不同时间不同数据样本的权重问题。面对时序数据问题，需要考虑不同时间的数据重要程度以及权重，对于越新的数据对模型的贡献越高，越远的数据对模型的贡献相应降低。

时序随机森林投票算法考虑模型中各决策树的数据时序特征，根据子数据集时间标签在总数据集时间标签的关系，进行加权运算。在运算时需要考虑整体样本数据集的范围，和子样本数据集的范围。用时序均值和时序方差刻画数据集的时序属性。

在磁盘故障预测模型中，数据将会以固定的频率进行采集，对于同一块磁盘其样本数据应该是连续的。但是由于磁盘数据采集时，可能由于网络抖动或者是服务器不稳定原因有少数样本丢失情况，导致数据不连续。所以在将数据集划分为子集时，同一数据子集存在不同的时间间隔情况。在数据时序分配策略中已经提出，本文将不同时间序列的数据分配到不同的训练、验证和测试集中，各集合的数据样本应该是时间连续和被排序后的数据。在随机森林投票算法中，将对每个数据集求出其数据样本的时间均值和方差，利用这两项数据对样本的时序性进行拟合。

为了解决此异常导致的数据时序不均衡问题，时序数据分配算法使用子集各时间标签的均值和方差拟合样本子集。其中时间标签的均值代表此子集在时间轴的大致位置，方差代表子集的分布范围。

设数据集样本为  $D = \{D_1, D_2, D_3, \dots, D_k\}$ ，其中  $k$  表示每个样本子集有  $k$  条数据， $D_i$  是第  $i$  个样本子集， $T_i$  为样本子集的时间标签向量；记  $T_i = \{t_{i,1}, t_{i,2}, t_{i,3}, \dots, t_{i,k}\}$ ，其中  $t_{i,j}$  为  $D_i$  样本子集的第  $j$  条数据  $d_{i,j}$  样本的时间标签。而样本子集  $i$  的均值和方差分别表示如式(3-2)与式(3-3)：

$$\bar{u}_i = \frac{1}{N} \sum_{k=1}^N t_{i,k} \quad (3-2)$$

$$\sigma_i^2 = \frac{1}{N} \sum_{k=1}^N (t_{i,k} - \bar{u}_i)^2 \quad (3-3)$$

记最新的时间样本标签的序号为  $k$ ，第一个时间标签样本序号为 1；整体数据样本的时间均值为  $\bar{u}$ ，方差为  $\sigma^2$ ，时序权重初始化算法的伪代码如下：

输入：随机森林中决策树的数目  $k$ ，记为  $T_1$  到  $T_k$ ；

输出：各决策树的初始权重  $W_i$ ；

Init:

$$\bar{u}_i = \frac{1}{N} \sum_{k=1}^N t_{i,k};$$

$$\sigma_i^2 = \frac{1}{N} \sum_{k=1}^N (t_{i,k} - \bar{u}_i)^2;$$

Begin:

For  $i \in [1, k]$ :

$W_i = 1$ ;

    If  $k == 1$ :

        Return  $W_i$

$$\chi = \frac{u_i - \bar{u}}{u_k - \bar{u}}; \lambda = \frac{1}{2}; \gamma = \frac{1}{2};$$

$$W_i = W_i * (\lambda * \chi + \gamma)$$

$$\chi = \frac{\sigma_i^2}{\sigma^2}, \lambda = \frac{1}{k}, \gamma = \frac{k-1}{k}$$

$$W_i = W_i * (\lambda * \chi + \gamma)$$

Return  $W_i$

在时序权重初始化算法中，我们使用样本子集  $i$  的均值  $\bar{u}_i$  和方差  $\sigma_i^2$  对时序特征进行刻画，按照子集对整体数据集贡献进行计算权值。我们使用均值和方差对  $W_i$  进行拟合，采用线性加权求  $W_i$ ，其中  $\lambda, \gamma$  为加权因子， $\lambda, \gamma \in [0, 1]$ ， $\lambda + \gamma = 1$ 。在第一次拟合权重时使用均值信息，第二次拟合时使用方差信息，各个数据集样本可以综合得到所有数据集样本均值和方差信息。根据时序数据分配算法可计算每个样本子集的初始权重，从而对随机森林的不同决策树进行初始权重的赋值。随机森林生成规则如下：

- (1) 确定随机森林模型的决策树个数  $k$ ，分别标号  $Tree_1$  到  $Tree_k$ ；
- (2) 将样本数据按照时间进行排序，分为  $k$  个区间，记为  $D_1$  到  $D_k$ ；



(3) 把数据对应到决策树，并且根据算法对决策树分配初始权重 $W_i$ ；

(4) 对随机森林中决策树进行训练；

随机森林中共由  $k$  个决策树组成，其中由数据远近分别赋予不同的初始权重。根据随机森林的生成规则，我们可以得到其逻辑架构如下图所示。

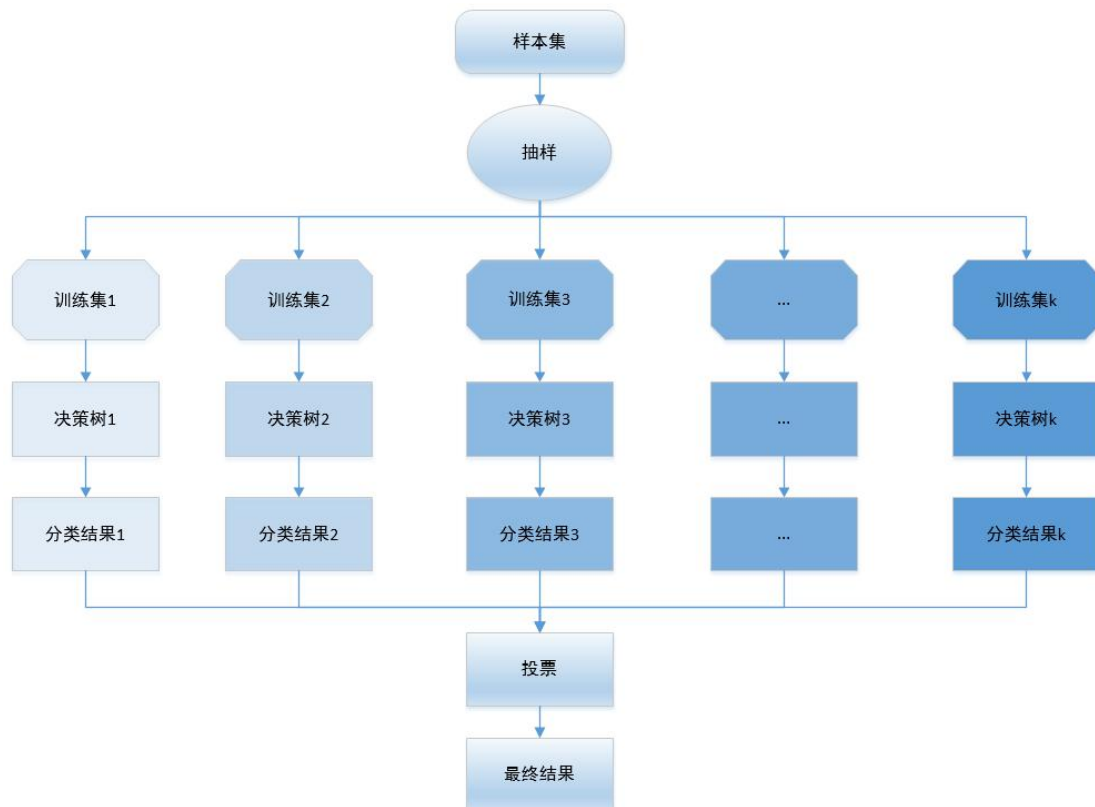


图 3-3 随机森林逻辑架构图

在样本集确定后，将数据抽样分发到训练集、验证集和测试集，然后将各数据按照时间顺序对应到各决策树中。根据数据集合的时序不同，将不同的数据集对应到每颗决策树中，同时使用时序权重初始化算法确定每棵树的始权重。

在模型建立后将会持续对线上的磁盘进行故障预测，但是随着时间不断的推移数据将累计越来越多，故障磁盘的模式可能发生变化，所以需要预测模型同样进行更新。在处理模型更新问题时主要有两个问题需要考虑，首先是最新周期数据加入训练模型中需要更新各决策的权重，其次是上周期预测后验数据处理。

## 3.2.2 负反馈更新模型

在每周期预测完成后，将会得到上周期的预测结果，此时将最新的数据与历史

预测数据重新进行训练。在本文的设计中，将随机森林预测模型中加入一颗后验决策树，将历史数据预测错误的部分加入到后验决策树中，同时需要更新各决策树的投票权重，对下一轮预测数据重新计算。

对于后验决策树，它作为负反馈模型中重要的一部分，负责对历史数据错误预测的部分进行再训练，期望能够重新学习错误数据特征。后验决策树的权重与上周期整体的预测结果有关，如果上周期的预测结果较差，说明坏盘模式修改程度更大，相应的后验决策树的地位更加重要，那么需要提升它的权重。反之说明坏盘模型修改较低，不需要后验的结果进行修正，从而它的权重较低。

对于随机森林的所有决策树而言，其权重随着每次的预测结果的不同而不断更新，从而可以保证决策树学习到最新的坏盘模式。对于某些预测较为准确的决策树而言，其权重应相应地进行增加，反之应该降低其预测权重。其具体算法如时序模型权重更新算法所示，其中包括了后验决策树权重的计算方法。

每棵树 $Tree_i$ 上周期数据预测准确率为 $P_i \in [0, 1]$ ，其更新模块的伪代码如下：

```
输入：随机森林中  $k$  颗决策树 $Tree_i$ 的初始权重 $W_i$   $i$  属于  $1, k$ 
输出：每颗决策树的最新权重 $W_i$ ，后验决策树的权重  $Post\_Tree$  的 $W^p$ 
Init:

$$\bar{P} = \frac{1}{k} \sum_{i=1}^k P_i$$


$$W_{init}^p = \frac{1}{k+1}$$

Begin:

$$\chi = \frac{P_j}{\bar{P}}, \lambda = \frac{1}{2}, \gamma = \frac{1}{2}$$


$$W_i = W_i * (\lambda * \chi + \gamma)$$


$$W^p = W^p * (\lambda * \bar{P} + \gamma)$$

Return  $W_i, W^p$ 
```

根据时序模型更新算法的描述，在普通决策树更新时考虑上周期所有决策树的预测情况，其中后验决策树的初始权重为所有树平均权重。模型更新具体流程如下：

- (1) 当前周期数据加入训练，根据时序数据分配算法求出数据集权重；
- (2) 将上周期预测错误的样本集重新输入到训练模型；
- (3) 参考历史周期各决策树预测结果，根据预测结果调整决策树权重；

在模型更新时需充分模型的泛化能力，在预测错误时将对应的决策树及时权重降低，在预测正确时将对应的决策树权重提升，本文使用时序模型权重更新算法进行随机森林各决策树权重的更新。

### 3.3 本章小结

本章介绍了数据处理和预测模型部分，提供了大规模磁盘故障预测的理论架构支持。将时序信息引入磁盘预测模型，提出时序随机森林的磁盘故障预测方法，包括数据时序特征处理和时序模型的优化。

数据时序特征处理部分，包括了数据时序分配策略和时序特征筛选算法，并在已有的基础上作出创新。提出数据时序分配策略，将时序数据按照时间动态分配到训练集、验证集与测试集。提出时序特征处理算法，将 S.M.A.R.T 属性的变化值拓展到模型特征中。

时序模型优化部分，描述了随机森林的各决策树投票算法，基于样本数据的时序特征，对随机森林中的不同决策树初始化不同权重，使用均值和方差进行时序数据的拟合。在随机森林中引入后验决策树提供负反馈更新，并引入决策树权重更新算法，动态学习错误预测数据，保证整体模型的准确率。

## 4 磁盘故障预测原型系统的设计与实现

本章在磁盘故障预测模型的基础上，增加了大规模数据中心的磁盘数据采集内容，以及巡检系统应用以及中心监控系统等方面内容。按照工业系统的角度进行说明，展现一个完备的磁盘故障预测应用与磁盘巡检的系统。本章将从整体系统的概述，包含各组成子系统，分别是磁盘数据采集系统，磁盘故障预测系统，磁盘巡检系统，中心监控系统。在各个子系统中，将会涉及到其部署、监控采集、应用等内容。

### 4.1 原型系统概述

为提升大规模存储中心存储节点可靠性，降低存储节点磁盘介质故障导致的数据丢失问题，我们利用故障预测系统进行提前故障预测，从而提前发现并修复坏盘。我们设计了一套大规模存储中心的磁盘故障预测原型系统，该系统由四个子系统组成：磁盘数据采集系统，磁盘故障预测系统，磁盘巡检系统，中心监控系统。其中磁盘数据采集系统完成磁盘数据采集，监控，容灾异地存储等功能，保证磁盘运行数据长期可靠存储；磁盘故障预测系统负责对磁盘数据进行清洗，对故障磁盘的预测告警功能；磁盘巡检系统负责对磁盘进行周期性地巡检，发现坏块并记录告知中心监控系统，同时接受来自磁盘故障预测系统的指令，进行对预测坏盘的巡检配置分发；监控中心系统负责对磁盘状态的监控，包括坏盘、坏块事件的响应，以及磁盘的 I/O 负载等。他们相互配合完成磁盘数据采集，故障预测以及磁盘巡检功能，他们的相互关系如图 4-1 所示。

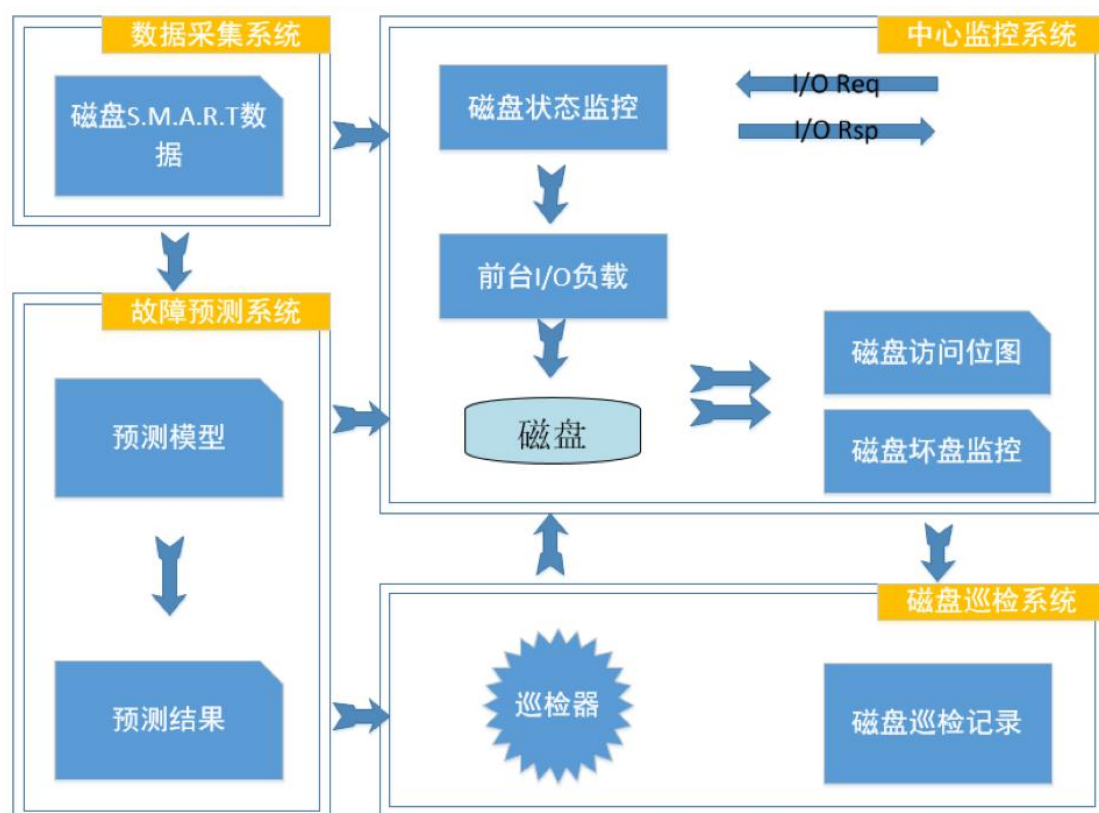


图 4-1 原型系统概览图

如图 4-1 所示，系统架构主要分为四部分，分别是磁盘数据采集系统，磁盘故障预测系统，磁盘巡检系统，中心监控系统。

数据采集系统主要完成磁盘 S.M.A.R.T 数据的采集任务。它首先在存储中心各存储节点部署采集程序，然后采集程序实时采集各存储节点的磁盘介质 S.M.A.R.T 数据。采集的磁盘 S.M.A.R.T 数据将发送到统一的网关进行数据转发，后台由至少两台服务器进行分发和网络数据容灾。最后数据将进行冷热存储，其中热数据将实时建立索引，保存在数据库中；冷数据将保存在数据仓库中，进行离线查询、归档。磁盘数据采集系统将监控每一台机器、每一块磁盘的状况，将异常信息及时同步到中心监控，并部署采集系统的告警模块，若分布式集群的存储节点有机器下线事件或分布式存储节点的磁盘出现异常、超时相应情况，系统将进行实时告警。分布式存储节点磁盘的异常包括该磁盘与分布式节点失去联系，磁盘命令响应超时，或者是读/写错误等情况。

故障预测系统首先通过磁盘数据采集系统读取历史的 S.M.A.R.T 数据，然后进行数据清洗来剔除异常数据，用特征工程规整数据，符合预测模型的数据输入类型。

故障预测系统在中心监控节点中得到故障磁盘的标签数据，对故障预测模型中的训练数据进行标签记录。在对故障磁盘预测的数据处理完毕后，将使用决策树模型对故障磁盘进行建模和预测。在中心监控节点中，运维人员参考历史的磁盘 S.M.A.R.T 数据进行磁盘检查，或者进行数据备份等操作，保障数据安全。在磁盘巡检系统中，磁盘巡检调度器将对故障磁盘进行巡检配置的修改、分发，从而可以尽快的找到磁盘坏块，从而提升分布式存储系统的可靠性。

磁盘巡检系统负责对存储介质进行定期巡检，发现潜在的磁盘坏块。磁盘巡检的周期或速率不同对磁盘的可靠性提升，和对前台的负载影响也不同，在巡检系统中，不同质量状况的磁盘巡检的参数不同。在磁盘故障预测系统中，被预测为坏的磁盘将发送到磁盘巡检系统，进行磁盘巡检配置的修改、更新。所以在分布性存储中心中，将会针对不同监控状况或者使用年限的磁盘，巡检系统制定不同的巡检参数，下发到不同的存储节点，应用到每一块磁盘上。对于预测为故障的磁盘，巡检系统将会加大对其检查粒度，从而期望更快地发现潜在扇区故障。对于巡检系统发现的坏块，将会上报到中心监控系统进行处理。

中心监控系统负责对磁盘的各项状态进行监控，还包括与运维人员的对接，以及各项告警系统的接入。其中磁盘的各项各项状态包括磁盘的 I/O 流量，磁盘的温度等各项 S.M.A.R.T 告警数据，以及磁盘的坏盘、坏块事件记录。在中心监控系统中，运维人员可以对磁盘的坏块进行检测并修复，或者是执行数据备份、换盘等操作。

该系统基于第 2 章、第 3 章的描述进行开发，包括后台管理控制进程和前台界面等。数据采集系统中部署的数据采集程序用 C++ 实现，采集完毕后数据传输服务由 Apache<sup>[71]</sup> 提供，Virtual IP 节点由 Nginx<sup>[72]</sup> 代理网关实现负载均衡，两个 Proxy 节点的数据告警业务由 Python 实现。采集系统中 S.M.A.R.T 热数据存储于 MYSQL 数据库，并在相关字段建立索引支持快速读取；S.M.A.R.T 冷数据存储于 EMR 数据仓库<sup>[73]</sup> 中，提供长期离线保存。

磁盘故障预测模型由 Python 的 Scikit-learn 函数库<sup>[74]</sup> 提供，支持决策树的生成调参等。模型的生成、定期更新等脚本命令也由 Python 实现，查询数据库数据并实时预测，将结果保存。前台网页由 Beego 框架搭建，HTML 实现数据展示，页面交互等功能。



## 4.2 磁盘数据采集系统

在磁盘故障预测中，最重要的部分为对磁盘进行建模，针对于不同类型或介质的磁盘需要进行建立不同的模型，而其中的第一步就是获取各种型号磁盘的运行数据。为获得大规模、高质量的磁盘历史运行数据，不断修正更新预测模型，我们需要对磁盘的运行数据进行实时采集并保存，为此我们设计了磁盘 S.M.A.R.T 数据采集系统。采集系统的框架如图 4-2 所示。

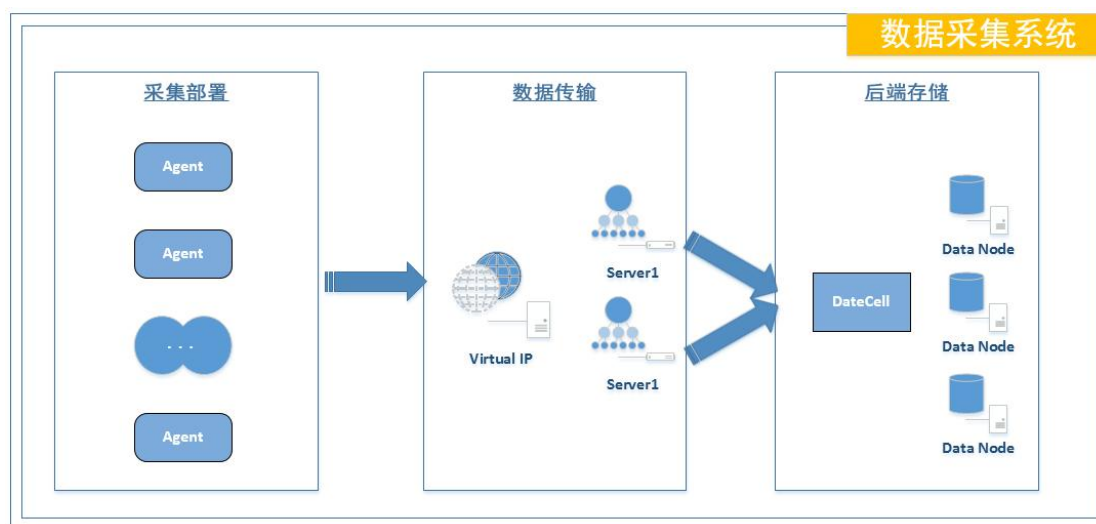


图 4-2 磁盘数据采集系统架构图

磁盘数据采集系统架构如图 4-2 所示，其中包含三个模块，相互配合完成磁盘数据采集功能。第一个模块为数据采集模块，在分布式存储节点中每台 Agent 有若干块磁盘提供数据存储服务。在部署磁盘数据采集程序时，将 S.M.A.R.T 采集程序部署在 Agent 上，采集程序将每小时对每一块磁盘采集一次 S.M.A.R.T 数据，然后将当前周期采集的数据发送到 Proxy 节点，由 Proxy 节点对数据进行分发到具体的一台 Server 进行处理，从而避免因单节点故障导致的磁盘 S.M.A.R.T 数据丢失。在多台 Server 对当前周期的数据进行校验，检查是否有存储节点宕机、磁盘下线等异常情况，如果有的话需将异常情况上报到中心监控节点，实现告警功能。在 Server 对当前周期数据校验完成后，将数据同时存到热数据区和冷数据区，其中热数据区提供磁盘故障模型训练、测试等工作，冷数据区用于对磁盘历史运行数据长期归档存储。

磁盘数据采集系统中部署在分布式存储节点的采集程序需采集多种数据，详细的数据名称和类型如表 4-1 所示。

表 4-1 磁盘数据采集程序数据名称

数据类别	数据名称	标记名称	数据类型
硬盘静态数据	服务器 IP	sever_ip	varchar64
	服务器型号	server_model	varchar64
	硬盘厂家	disk_vendor	varchar64
	硬盘型号	disk_model	varchar64
	硬盘序列号	disk_sn	varchar64
	磁盘盘符	disk_dev	varchar64
	硬盘容量	disk_size	int64
硬盘动态数据	磁盘 S.M.A.R.T 数据	disk_S.M.A.R.T	varchar1024
时间戳	采集时间	time	datetime

根据磁盘故障预测系统中，故障磁盘训练模型的需要，我们将采集的数据分为三个维度进行存档。首先是磁盘静态数据，我们将记录存储服务器节点配置信息和存储磁盘介质配置信息两部分的数据。在存储服务器节点中，我们将采集服务器地址（sever\_ip）数据和服务器型号（server\_model）数据，进行定位到服务器级别；在存储磁盘介质中，磁盘数据采集程序记录磁盘生产厂家（disk\_vendor），磁盘型号（disk\_model），磁盘序列号（disk\_sn），以及磁盘容量（disk\_size）和磁盘所在盘符（disk\_dev）数据，其中磁盘容量为 int64 型，单位是 GB。第二部分是磁盘运行时产生的 S.M.A.R.T 数据，采集程序将自动采集当前磁盘记录的前 25 个 S.M.A.R.T 属性，包括属性中的有效的三个值：rawvalue, worstvalue, value。磁盘动态运行的 S.M.A.R.T 一般不会超过 30 个，属性标号靠后的数据对磁盘故障预测功能的影响不大。第三部分是采集磁盘数据那一刻的时间戳，用于在训练或者预测的时候标识数据产生的时间。采集程序将采集的完毕的数据形成标准化文本文件，图 4-3 展示了一次采集的数据记录。



```
server_ip:10.174.234.140
server_model:TS8
disk_vendor:WDC
disk_model:WD2000FYYZ-88UL1B0
disk_sn:WD-WMC1P0FADWVM
disk_dev:/dev/sdb
disk_size:2048

smart:1_cur:100,1_worst:100,1_rawvalue:0,1_flag:11,2
_cur:138,2_worst:138,2_rawvalue:76,2_flag:5,3_cur:16
9,3_worst:169,3_rawvalue:38678757747,3_flag:7,4_cur:
100,4_worst:100,4_rawvalue:13,4_flag:18,5_cur:100,5_
worst:100,5_rawvalue:0,5_flag:51,7_cur:100,7_worst:1
00,7_rawvalue:0,7_flag:11,8_cur:145,8_worst:145,8_ra
wvalue:24,8_flag:5,9_cur:98,9_worst:98,9_rawvalue:18
787,9_flag:18,10_cur:100,10_worst:100,10_rawvalue:0,
10_flag:19,12_cur:100,12_worst:100,12_rawvalue:13,12
_flag:50,192_cur:100,192_worst:100,192_rawvalue:13,1
92_flag:50,193_cur:100,193_worst:100,193_rawvalue:13
,193_flag:18,194_cur:193,194_worst:193,194_rawvalue:
146030395423,194_flag:2,196_cur:100,196_worst:100,19
6_rawvalue:0,196_flag:50,197_cur:100,197_worst:100,1
97_rawvalue:0,197_flag:34,198_cur:100,198_worst:100,
198_rawvalue:0,198_flag:8,199_cur:200,199_worst:200,
199_rawvalue:0,199_flag:10

time:2018/07/03 15:02:40
```

图 4-3 一次采集数据记录

图 4-3 展示了一次采集程序采集的数据记录，其中包括三个维度共九项数据字段。其中磁盘静态数据维度中服务器信息部分展示了磁盘的服务器 IP 和服务器型号数据，显示本次磁盘 S.M.A.R.T 数据来自于 IP 为“10.174.234.140”的一台“TS8”存储节点。磁盘静态数据维度中磁盘信息部分展示了其他五项数据，显示本次数据来源于“WDC”厂家生产的型号为“WD2000FYYZ-88UL1B0”的磁盘，其磁盘的序列号为“WD-WMC1P0FADWVM”，并且位于服务器的“/sdb”盘符，这块盘的容量为 2048GB。

## 4.3 故障预测系统

在数据采集系统中，海量的磁盘运行数据源源不断的被收集，存到后台的存储平台供故障预测系统建模、预测。在磁盘故障预测系统中，首先对 S.M.A.R.T 数据进行数据清洗，后对故障磁盘数据进行建模，判断磁盘在接下来的一段时间是否会发生损坏，同样我们也需要不断地根据下一周期的数据进行模型更新，确保该模型

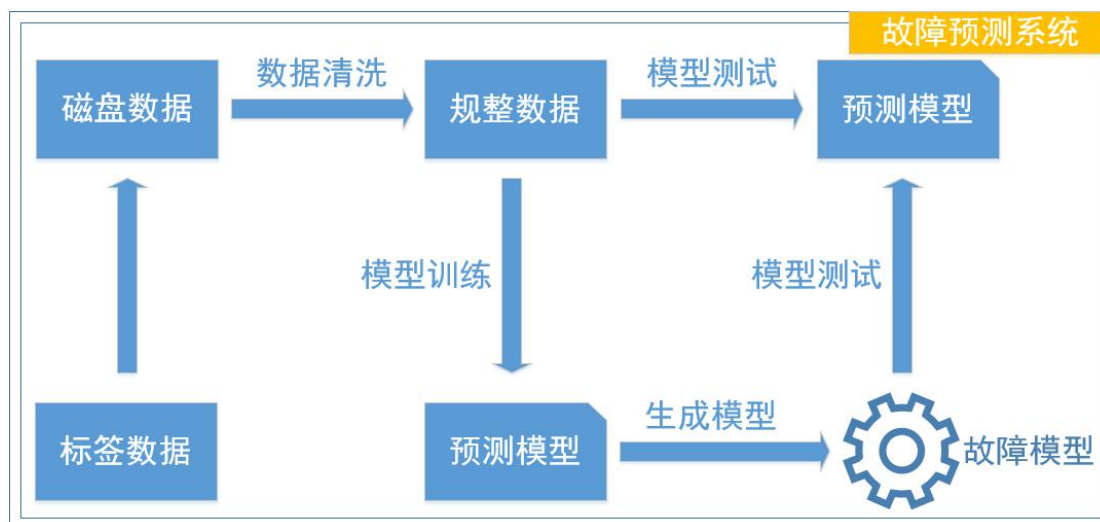


图 4-4 磁盘故障预测逻辑图

磁盘采集系统采集的磁盘数据读取后，从中心监控系统中取到坏盘的标签数据，把标签数据与磁盘 S.M.A.R.T 数据作为磁盘的特征数据集，在特征处理后得到规整的训练和测试数据。到模型训练阶段时，规整的训练数据经过模型训练，学习坏盘数据特征生成故障模型。通过规整的磁盘数据测试，得到当前周期的坏盘。通过这样的训练、生成模型、测试的步骤，最终对每个周期的磁盘数据进行训练。并且把最新周期的预测结果保存数据到前端，将当前周期磁盘预测结果保留进行后验，进行模型的修正。在每次的迭代中，我们依据上周期模型预测错误记录，进行模型地更新调整。

在故障磁盘的训练环节，将磁盘的历史数据统一处理，按照第四章提供的特征选择方式将磁盘数据格式化为共 19 个 S.M.A.R.T 复合属性。并且将故障盘标签数据集中的磁盘标签置为 1，将非故障盘的标签置为 0，进行坏盘的标识。通过决策树算法进行训练，并将当前周期的模型保存，进行最新周期的坏盘预测。图 4-5 展示了被规整的磁盘数据。

```
[2048,
76, 96, 100, 93, 90,
100, 100, 28, 3, 100,
0, 0, 3, 0, 0,
1, 0, 0, 0
1]
```

```
[1024,
68, 96, 100, 94, 90,
100, 77, 28, 64, 100,
0, 62, 3, 50, 0,
1, 3, 0, 0
0]
```

(a) 某坏盘样本数据

(b) 某好盘样本数据

图 4-5 被规整的数据

如图 4-5 所示，磁盘被规整后的属性包括三部分，第一部分为磁盘的基本数据，在训练时仅保存磁盘的大小一项，第二部分为磁盘的动态 S.M.A.R.T 属性，其中包括 19 个被处理后的 S.M.A.R.T 属性。第三部分为数据的标签，标识这条数据是好盘还是坏盘。如图(a)所示，磁盘的大小为 2048GB，磁盘共有 19 维 S.M.A.R.T 属性，最后一维是磁盘的标签数据，标识这是一块坏盘；图(b)展示的是一块 1024GB 的好盘。

在坏盘的预测阶段，首先利用当前周期的训练数据对模型训练，从而得到坏盘特征。然后利用第三章数据处理的方法对测试数据进行特征工程等处理，形成规整的测试数据，最后使用训练好的模型对规整的测试数据进行预测，来判断该磁盘是否会在接下来的一段时间发生故障。为了能够正确进行预测，我们需要将最新周期的磁盘数据进行同样的特征处理，归一化为同样的训练数据格式。

在模型更新阶段，需要利用上一阶段的故障磁盘预测结果，和正确的故障标签数据进行后验，来帮助对现有模型的修正。其过程与第三章中模型更新部分相同，利用上一阶段预测错误的磁盘，加上正确的标签作为增加的数据集，并且配以更高的权重，从而增强错误数据预测的学习。利用上周期预测的结果的后验数据，进行模型的更新可以帮助我们的模型可以追踪到磁盘的最新模式，从而提高磁盘故障预测的准确率。

## 4.4 磁盘巡检系统

磁盘巡检系统对存储中心的存储节点磁盘进行周期性的巡检，发现潜在的磁盘扇区故障，并且告知运维人员进行修复。它负责对巡检指令的下发，巡检配置的修改、更新。根据中心监控系统记录的不同监控状况的磁盘，设置不同的巡检配置，下发到对应的存储节点磁盘介质上，从而期望更快地发现故障的磁盘扇区。其系统的逻辑架构如图 4-6 所示。

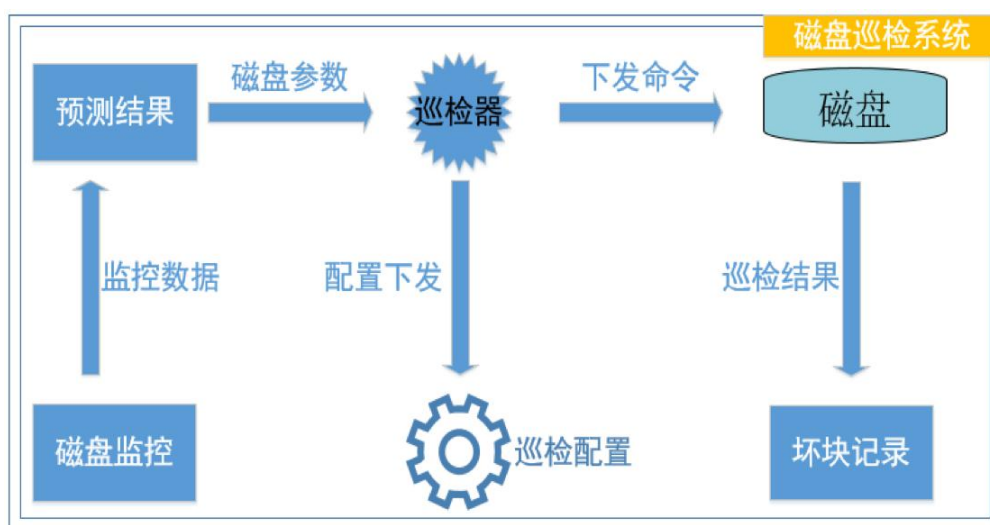


图 4-6 磁盘巡检系统架构图

如图 4-6 所示，磁盘巡检系统共分为三个部分：第一部分是磁盘参数获取，其中包括由中心监控系统提供的磁盘运行参数、I/O 负载等，以及磁盘故障预测中坏盘的预测结果；第二部分是磁盘巡检配置的修改、更新，以及磁盘巡检控制命令的下发；第三部分是将巡检的结果记录，供运维人员进行修复或者数据备份。

在当前周期磁盘故障预测完成后，将预测结果发送到磁盘巡检模块中，修改、更新巡检参数，并及时应用到当前周期的巡检。对于预测为坏盘的磁盘，加快巡检速度，缩短巡检周期。对于预测为正常的磁盘，巡检参数不进行特殊处理。对于多次预测为坏盘的磁盘进行数据迁移，修改存储系统配置不进行数据写入。

磁盘巡检的相关参数有巡检周期，巡检时间间隔，巡检当前位置等配置项。在工程实现中分为静态配置和动态配置：静态配置文件记录了存储节点开关，两次巡检命令的下发时间间隔，巡检下发阈值等参数；动态巡检配置包括每一块磁盘的配置信息，其中包括磁盘的序列号，已完成巡检的次数，和巡检的开始时间、最后一次巡检时间，当前的巡检进度等。配置文件用 xml 标记语言记录，其具体的配置项数据如图 4-7 所示。



```
<root>
  <conf>
    switchon          = 1
    sleep             = 100
    sync_time         = 3
    period             = 60
    read_most         = 100
    write_most        = 100
    util_most         = 70
    segment_default   = 12800
  </conf>

  <disk>
    <sdb>
      diskswitch      = 1
      segment_now     = 12800
    </sdb>
    ...
    <sdm>
      diskswitch      = 1
      xsegment_now    = 6400
    </sdm>
  </disk>
</root>
```

图 4-7 巡检系统的静态配置文件

如配置文件图 4-7 所示，巡检的静态文件包括两部分，第一部分是存储节点的整体配置文件，用于控制整个存储节点所有磁盘的巡检进程。第二部分是单块磁盘的配置，用于对更细粒度的磁盘级别的巡检控制。

在第一部分中，其中 `switchon` 标识巡检的开关，1 为巡检开启，0 为巡检关闭；`sleep` 配置项为两次巡检命令下发的时间间隔，此处的 100 指两次巡检命令下发的时间间隔为 100ms；`sync_time` 配置项为配置文件由内存保存到文件的频率，次数为 3 小时一次；`period` 配置项为巡检的周期，单位是天；接下来的三项配置 `read_most`，`write_most`，`util_most` 为巡检下发的阈值，当磁盘的负载都低于这三者时才可以进行巡检操作，以免对系统前台的影响多大；`segment_default` 配置项为默认的巡检长度，后续可根据磁盘的运行状况进行调整。

在第二部分中，用 `disk` 标签栏进行区分，其子目录是每一块被巡检的磁盘，用

“sdb”、“sdc”等进行标识。每一块磁盘有两个配置项，分别是巡检的开关，和单次巡检的长度。diskswitch 配置项代表这该块磁盘的巡检是否开启，其中 1 是标识开启，0 标识关闭；segment\_now 配置项为单次巡检下发的长度，当磁盘被预测为坏盘时该值将相应地增加，其单位是扇区个数，如 6400 指一次巡检命令下发 6400 磁盘扇区。

```
<root>
  <sdb>
    sn          = WD-WMC1P0FADWVM
    loops_count = 1
    finished     = 0
    start_time  = 2019-01-03_20:24:01
    last_time   = 2019-03-05_19:45:28
    lba_now     = 1284915200
    lba_max     = 3907029168
  </sdb>
  ...
  <sdm>
    sn          = WD-WMC1P0F62X62
    loops_count = 1
    finished     = 0
    start_time  = 2019-01-03_20:24:01
    last_time   = 2019-03-05_19:45:28
    lba_now     = 1287232000
    lba_max     = 3907029168
  </sdm>
</root>
```

图 4-8 巡检系统的动态配置文件

如图 4-8 所示，巡检的动态配置文件由每一块磁盘的配置项组成，其中共有 7 项配置项对当前磁盘的巡检状态进行记录。sn 配置项代表磁盘的序列号，是磁盘唯一的标识；loops\_count 配置项为磁盘巡检已经巡检完毕的周期计数，此处的 1 指磁盘已完成一次全盘巡检；finished 配置项标识当前磁盘是否已经提前完成了巡检任务。配置项 start\_time 和 last\_time 分别标识磁盘开始巡检的时间和最后一次巡检的时间，用于对磁盘巡检任务的查询跟踪；配置项 lba\_now 和 lba\_max 分别标识巡检的当前位置和需要巡检的量，从该配置项可对巡检的进度进行跟踪。

## 4.5 中心监控系统

中心监控系统负责各个子系统之间的数据监控和异常上报，以及告警运维手动处理等工作。中心监控系统对接数据采集系统、磁盘故障预测系统、磁盘巡检系统以及监控各个模块的数据异常，对异常数据及时告警并在监控系统上告警。如图 4-9 为监控系统逻辑架构。

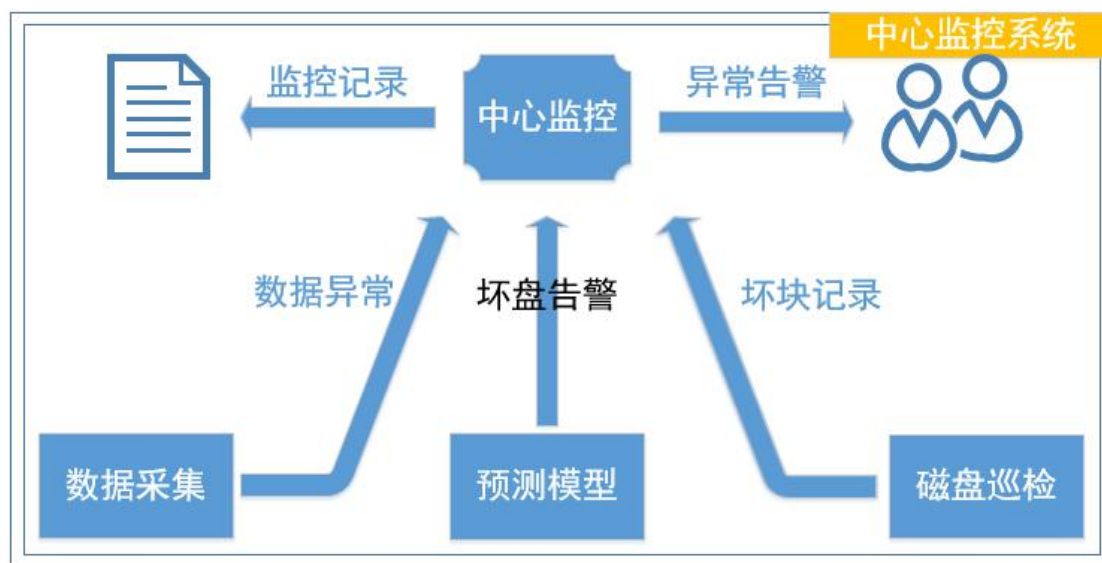


图 4-9 监控系统逻辑架构图

中心监控系统主要分为三个模块，另外承担告警运维与记录监控记录等工作。三个模块分别是数据采集系统，故障预测系统，磁盘巡检系统，在子系统出现异常时，将会告警到运维人员，并且将在系统进行告警。

### 4.5.1 数据采集系统监控

磁盘 S.M.A.R.T 数据采集系统有多个模块接入中心监控系统，主要包括存储节点异常信息，采集 S.M.A.R.T 历史信息。其中节点的异常信息包括存储节点的宕机异常，采集 S.M.A.R.T 磁盘的坏盘监控，磁盘更换等。其具体的监控上报如图 4-10 所示。



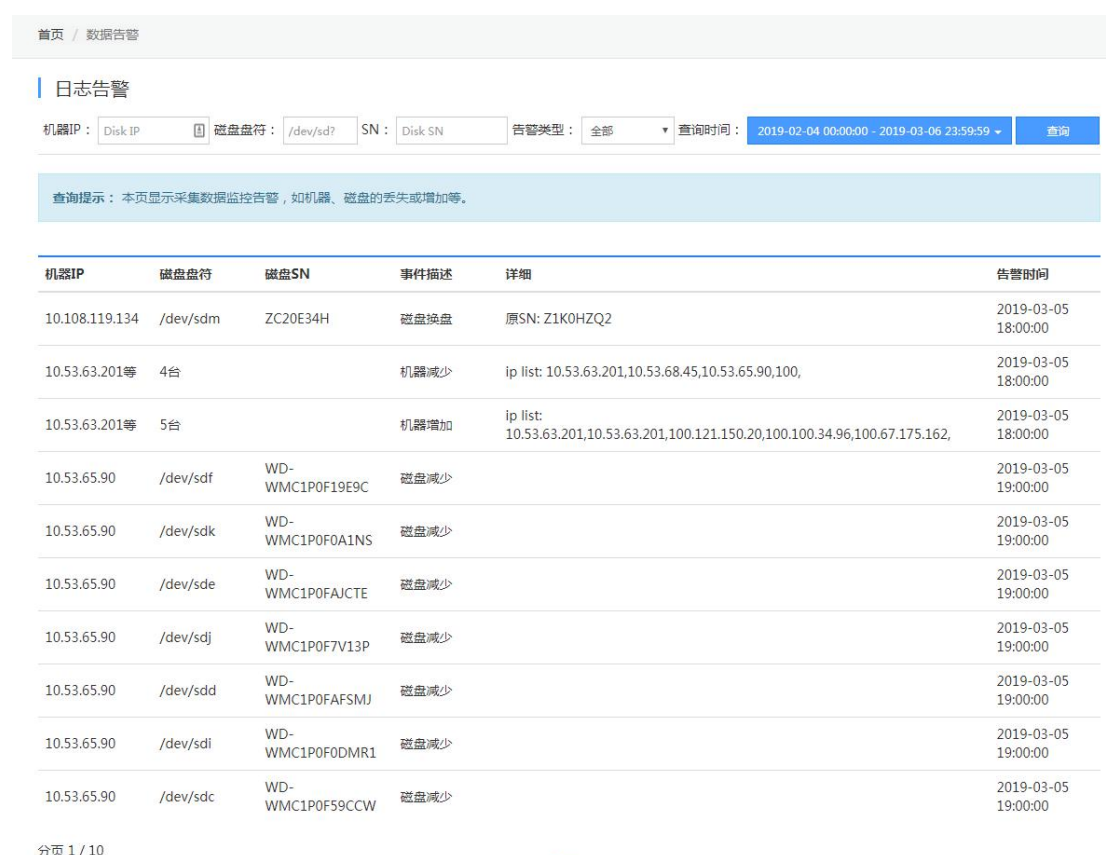


图 4-10 中心监控系统监控数据

如图 4-10 展示了中心监控系统对磁盘数据采集模块中的异常监控, 它监控磁盘换盘、存储节点的增删事件, 同时记录告警时间以及监控机型等。运维人员可通过监控系统定位到具体的存储节点和磁盘, 进行修复工作。

磁盘 S.M.A.R.T 历史数据定位可辅助运维人员在磁盘故障时进行诊断, 提升采集系统运维水平, 从而提高存储系统的可靠性。如图 4-11 所示, 运维人员可以从中心监控系统查询磁盘历史 S.M.A.R.T 数据信息, 在对磁盘运行状态不确定时辅助决策, 对预测故障的磁盘进行换盘或是数据备份操作。



图 4-11 中心监控系统 S.M.A.R.T 数据查询

## 4.5.2 故障预测模型监控

中心监控系统可监控磁盘故障预测的模型情况，支持在监控系统中进行模型的手动更新和模型质量的查看。可实时追踪到模型的最新周期质量情况，供运维人员参考进行磁盘修复或数据备份操作，巡检系统可参考预测结果进行巡检配置的调整和更新。



图 4-12 中心监控系统模型展示

如图 4-12 所示，是磁盘故障预测模型的测试模型展示页面，左侧为模型实时训练模块，在系统中可进行在线生成模型并且应用到线上系统，并实时展示模型的预测效果等。右侧为线上数据预测模块，可实时展示最新周期的预测结果，包括预测的准确率、召回率、误报率等数据，辅助运维人员和巡检系统进行决策判断。

磁盘故障预测模型将每周期进行更新，同时会对当前周期的坏盘进行预测，预测的结果将会同步到巡检系统对磁盘的巡检配置进行调整，同样也上报到中心监控系统，并在首页展示。如图 4-13 为磁盘故障预测系统首页，展示了系统的监控主面板和最新周期的故障磁盘预测数据。

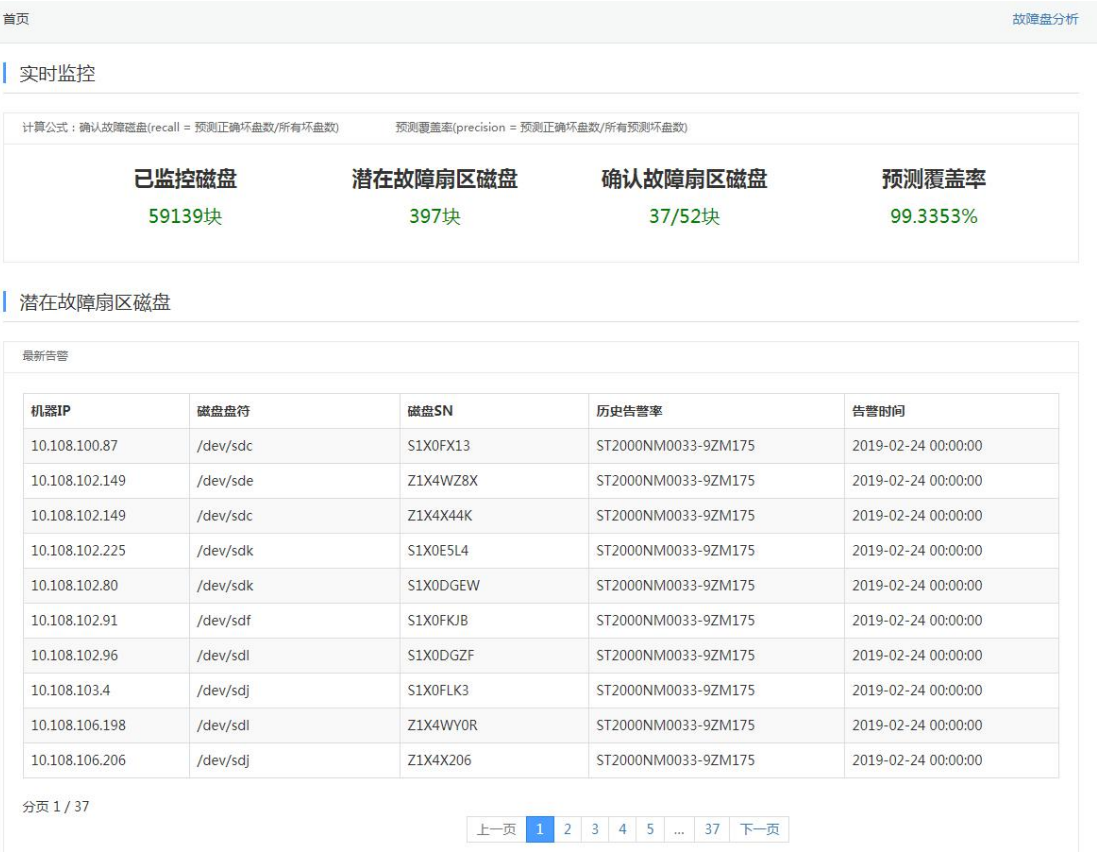


图 4-13 中心监控系统故障磁盘预测结果

如图 4-13 所示，是中心监控系统的首页界面，上方展示了磁盘故障预测的面板数据，下方为最新周期的故障磁盘预测信息。面板数据分别是已监控磁盘的数目，潜在故障磁盘的数目，以及确认故障磁盘的数目等，方便运维人员对磁盘故障预测系统准确率的掌握。下方为当前周期预测的故障磁盘，此数据供巡检系统调整磁盘的巡检速率，更快地发现磁盘坏块。

## 4.5.3 巡检系统的监控

中心监控系统对磁盘巡检模块进行监控，检测存储系统中存储节点的磁盘巡检状态，磁盘巡检的进度和巡检到的坏块等数据，及时知会运维人员进行修复。磁盘巡检模块运行时，监控系统会持续监控磁盘的负载、I/O 等情况来确保磁盘巡检时不会处于高负债状态。监控进程记录所有磁盘的坏块数据，并将对其进行分类统计，在磁盘坏块层面对存储系统的可靠性分析。

监控系统持续地监控磁盘的各种运行参数，并且上报到集中管理平台中，辅助磁盘巡检模块工作。在磁盘巡检命令下发时，需要获取磁盘的运行负载，当磁盘的负载过高时，说明磁盘正处于负荷较高的 I/O 操作，此时巡检过程应暂停，等磁盘负载较低时再进行下发。另外，中心监控系统负责设置磁盘写入的开关，在磁盘发生故障或者多次预测为坏盘时，告警系统将会以告警的方式知会运维人员进行处理，同时关闭数据写入开关，避免数据写入导致数据丢失。在磁盘初次预测为坏盘时，将会以页面告警的形式在系统中进行记录，调整巡检的速率进行错误的提前发现。

已发生故障的磁盘坏块数据，监控系统将对其进行分类统计，从系统的角度优化故障分析模型。对于在分类统计模块中，支持对磁盘坏块按型号进行数据统计，支持按型号统计故障百分比等。该功能在磁盘型号的层面上对磁盘坏块分析，辅助运维进行决策，同时将更新故障模型中对不同型号磁盘的预测权重，相关统计结果如图 4-14 所示。

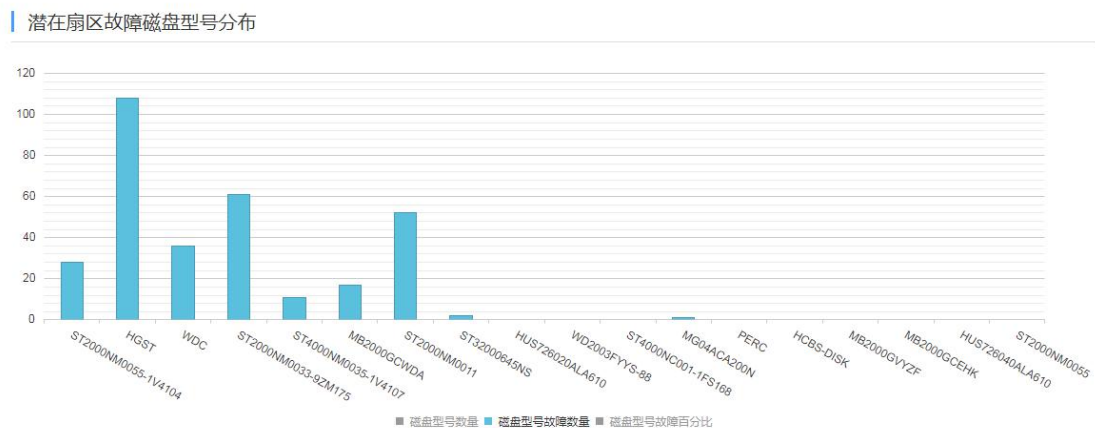


图 4-14 中心监控系统磁盘坏块统计

如图 4-14 所示为中心监控系统坏盘统计，图片中为按磁盘型号进行故障统计，

可见“HGST”型号磁盘坏块数目最多，其次为“ST2000NM0033-9ZM175”，可见磁盘坏块集中分布在少数的几个型号中，该分布与数据中心中磁盘型号较为集中有关。

### 4.6 本章小结

本章为数据中心建立了一套智能化故障预测应用原型系统。原型系统由磁盘数据采集系统，磁盘故障预测系统，磁盘巡检系统和中心监控系统共四个子系统组成。同时详细地介绍了大规模数据中心的原型系统各个模块，以及各个模块的功能、架构、联系等，并且给出了相应的数据结构以及对外的接口实现。

四个子系统相互配合保证数据中心存储介质的可靠性，提升存储中心智能化水平。数据采集系统为存储中心积累数据价值，供故障预测系统进行训练预测；故障预测系统提前预测磁盘故障，提升巡检系统价值，辅助进行坏盘诊断；磁盘巡检系统进行坏盘扫描，处理磁盘坏块从而提升存储系统的可用性；中心监控系统为三个子系统提供全局监控功能，联合运维人员增强系统的可靠性。

在实际应用中，大规模的数据中心在数据采集等方面会出现各种意料之外的情况，比如由于网络抖动带来的数据异常，或者机器重启等原因导致的磁盘数据丢失等，数据中心设计类似的模块时，需要建立对应的监控系统保证其运行质量。

## 5 系统分析测试

本章为系统分析测试，主要对磁盘故障预测系统与磁盘巡检的功能与性能进行验证和测算。其中包括实验环境描述，数据集来源与数据划分，故障预测与磁盘巡检的功能测试，评价指标与性能测试。其中各测试模块与评价指标描述中，按照故障预测与磁盘巡检分开讨论。

### 5.1 实验环境

磁盘故障预测在实验过程中需要大量的计算资源，为提升模型训练效率，本文所有的实验在腾讯云服务器<sup>[75]</sup>上完成。服务器版本是 Linux 3.10.107，物理内存是 128GB，存储为 500GB 物理硬盘，配备 2 个物理 CPU，型号为英特尔至强 E5-2620，主频 2.10GHz。此外，本文使用 Scikit-learn 开源机器学习库，模型开发语言为 Python。决策树模型使用 Python 建立并调参，特征选择与数据处理方式如第三章数据处理方式。

### 5.2 数据集与特征选择

#### 5.2.1 数据源描述

本章测试使用的数据集有两个来源，第一个是某数据中心的存储节点（记为 Company-T），采集时间为 2017 年 9 月 27 至 2018 年 7 月 30 日；共计 7 个型号的 64063 块企业级 HDD 磁盘，其中坏盘为 409 块，采集频率为每小时一次，共计约 4.8 亿条带时间戳数据记录。第二个数据来源是 Backblaze<sup>[76]</sup>，采集时间为 2013 年 4 月 10 日至 2018 年 6 月 30 日；挑选其中数据量较多的 16 个型号的 140671 块企业级 HDD 磁盘，其中坏盘为 6609 块，其数据采集频率为一天一次；各数据集如下表。

表 5-1 Company-T 数据集按型号分布

Company-T 磁盘型号	好盘数目	坏盘数目
ST2000NM0055-1V4104	25145	153
WDC	8265	43



## 华中科技大学硕士学位论文

ST2000NM0033-9ZM175	6880	37
ST4000NM0035-1V4107	3632	24
MB2000GCWDA	1079	15
ST2000NM0011	516	8
总计	63654	409

表 5-2 Backblaze 数据集按型号分布

backblaze 磁盘型号	好盘数目	坏盘数目
ST4000DM000	37006	3157
ST12000NM0007	21727	106
HGST HMS5C4040BLE640	16326	171
ST8000NM0055	14557	140
ST8000DM002	10067	187
HGST HMS5C4040ALE640	8714	145
Hitachi HDS722020ALA330	4774	225
Hitachi HDS5C3030ALA630	4664	141
ST3000DM001	4706	1591
Hitachi HDS5C4040ALE630	2719	82
ST31500541AS	2188	346
ST6000DX000	1938	64
WDC WD30EFRX	1329	171
ST10000NM0086	1223	3
HGST HUH728080ALE600	1076	8
Hitachi HDS723030ALA640	1048	72
总计	134062	6609

如表 5-1 与表 5-2 所示，数据集有两个明显趋势，一是磁盘型号众多，给故障磁盘建模增加难度；二是各型号磁盘中坏盘的比例较少，建模时将会存在样本不均匀问题，下一小结将说明如何处理模型训练时样本不均匀问题。



## 5.2.2 数据集划分

在磁盘故障预测系统中，由于磁盘型号众多且不同磁盘生产商的不同型号磁盘的 S.M.A.R.T 属性值计算公式可能有所不同，为了减少因为厂商、型号所带来的影响，我们在两种数据源中各自选择其中磁盘数量最多的三种型号作为代表，将其整体作为数据集进行试验验证。其中某数据中心磁盘型号 ST2000NM0055-1V4104 记为数据集 T-1，磁盘型号 WDC WD2000FYZZ-88UL1B0 记为数据集 T-2，磁盘型号 ST2000NM0033-9ZM175 记为数据集 T-3。在 Backblaze 数据集中，磁盘型号 ST4000DM000 记为数据集 B-1，磁盘型号 ST12000NM0007 记为数据集 B-2，磁盘型号 HGST HMS5C4040BLE640 记为数据集 B-3。

磁盘故障预测中，坏盘的故障率约为每年千分之三，且坏盘会被马上下架，磁盘正负样本分布极不均衡。为了平衡正负样本比，我们按照坏盘产生的时间分布对各数据集进行统计，最终选取坏盘发生较为密集的一段时间周期的数据，来进行磁盘故障预测的训练和测试。如图 5-1 所示了 T-2 磁盘的坏盘发现时间分布关系。

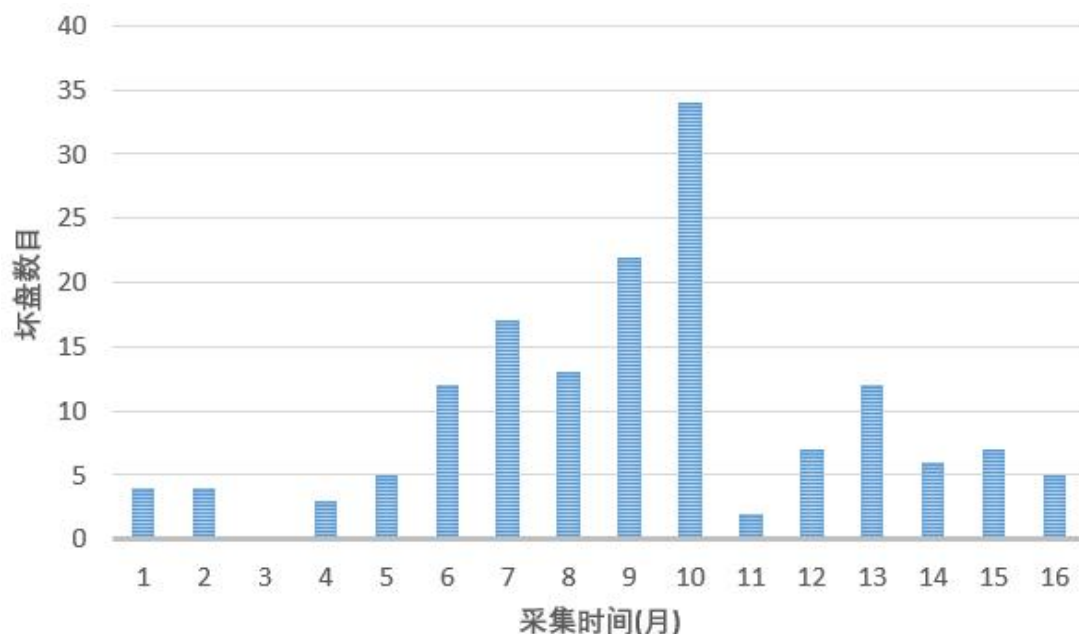


图 5-1 T-2 数据集坏盘发现时间分布

如图 5-1 为坏盘分布-时间柱状图，横轴是时间坐标轴，纵轴是每月的坏盘产生数。可见坏盘产生数目较少且较为分散，我们对 T-2 数据集选择了第 6 至第 10 月的数据集，作为磁盘故障预测模型的训练和测试。对于其他的数据集我们采用类似方

式，筛选出坏盘数目较为密集的若干周期的数据。

在进行上述时间选择后，在一定程度上缓和了正负样本不均衡的问题，但是负样本比例仍然相对较低。由于数据的不均衡性对模型的预测性能影响较大，最终模型不能有效地学习到正样本相关模式，所以需要均衡正负样本比。我们首先将磁盘进行统计分析，划分为正负样本。然后依据正样本的数值，随机删除一些负样本的数据，使得正负样本比约为 50: 1。具体的数据如下表所示，由于系统采集不稳定或网络抖动原因，部分 Samples 可能有所缺失。

表 5-3 各数据集的政府样本比例

数据源	磁盘型号	负样本数	正样本数
tencent-1	ST2000NM0055-1V4104	174787	3669
tencent-2	WD2000FYYZ-88UL1B0	4957	1032
tencent-3	ST2000NM0033-9ZM175	4428	883
backblaze-1	ST4000DM000	1458462	28762
backblaze-2	ST12000NM0007	43273	890
backblaze-3	HGST HMS5C4040BLE640	8760	1708
总计		1694667	36944

如表 5-3 所示，某数据中心与 Backblaze 共 6 种磁盘型号组成的数据集，在样本均衡时随机去掉了部分正样本，最后保证正负样本比约为 1 比 50。在预测时对同一块磁盘的多个样本，如有一个样本预测为坏的话，那么就认定此盘为坏盘。对于坏盘的认定，本文采取的方式是如果磁盘在接下来 7 天内损坏，则认定为坏盘。下面将使用上述 6 种型号组成的数据集进行训练与测试。

## 5.3 功能测试

本节将对对比系统设计的评估指标，对系统各模块的功能进行逐项测试。在内容组织上，根据磁盘故障预测系统和磁盘巡检系统两个部分，分别进行性能评估测试。

### 5.3.1 磁盘故障预测系统测试

在磁盘故障预测系统中，将分磁盘故障模型训练和测试模块两部分，分别进行

功能完整性测试。在故障磁盘模型训练模块中，首先需要对历史数据进行读取，然后预处理后进行特征选择，最后整合到决策树模型中进行学习训练。在数据预处理后，按照第二章的方法进行特征处理，共形成 19 维的数据向量与其他附属属性，如磁盘样本标签数据和磁盘容量属性。

在故障磁盘训练完成后，将会生成一颗当前周期使用的决策树模型，测试数据将会使用当前周期生成的决策树模型进行预测输出，预测当前周期可能会坏掉的磁盘，从而完成对一块磁盘的预测功能。生成的决策树模型可视化简化后如图 5-2 所示。

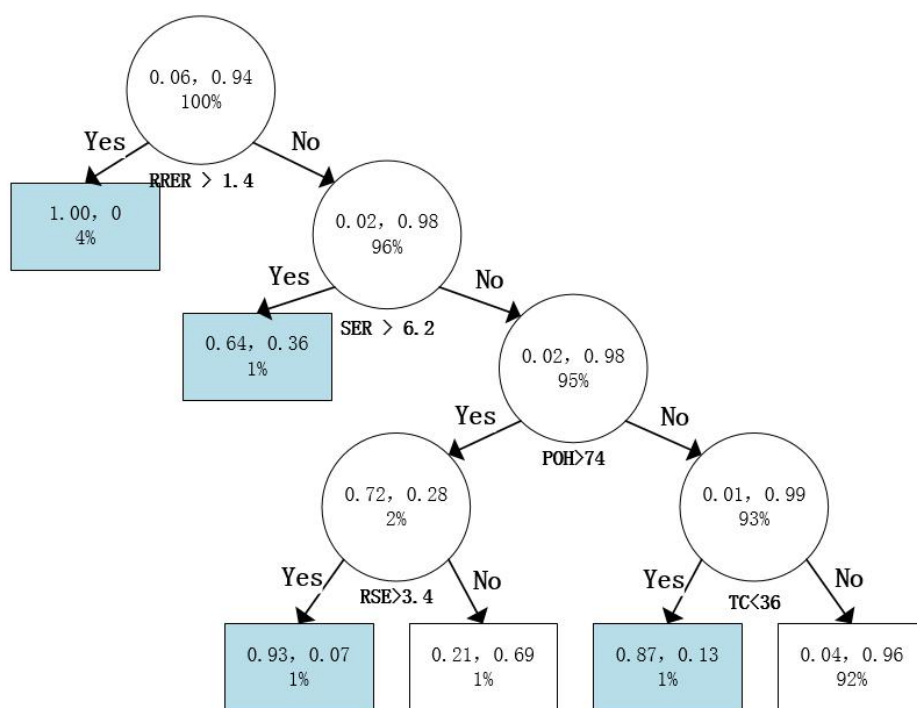


图 5-2 决策树模型可视化图像

在当前预测周期结束后，将会把决策树保存在本地，进行故障磁盘的预测工作。如上图所示，是一颗简化的被训练后的决策树模型，在最上方的分支节点是决策树的第一个判断分支。数据到这个节点共有 6% 的正样本数据，和 94% 的负样本数据，该节点使用 RPER 属性进行判别，如果大于 1.4 那么将会直接判别为坏盘，否则进入下一个节点。对于判别为坏的节点，其中全部是正确的分类，错误的分类数目为 0。对于下一步继续判别的节点，分别包括有 2% 的坏盘与 98% 的好盘，其数据量共包括所有数据的 95%。依次递归进行下一次判断，直到所有的节点分类结束或者是决策

到达预先设定的剪枝条件。

故障磁盘预测是一个二分类问题，将会对每块磁盘根据当前周期的所有采集的 S.M.A.R.T 数据进行评估预测，最终对每一块磁盘将会产生一个预测为坏盘的 0 到 1 的概率，数值越大表示当前磁盘越有可能在下个周期产生损坏，表 5-4 展示了一次故障磁盘的预测结果。

表 5-4 预测结果展示

磁盘序列号	预测概率	参考结果
ZC20LBN0	0.05	好盘
ZC20LBPH	0.03	好盘
ZC20LBS5	0.01	好盘
ZC20LBNH	0.13	好盘
<b>ZC20LC37</b>	<b>0.79</b>	<b>坏盘</b>
ZC20L9GR	0.04	好盘
<b>ZC20KKZA</b>	<b>0.88</b>	<b>坏盘</b>
ZC20LBSX	0.21	好盘
<b>ZC20LCGY</b>	<b>0.99</b>	<b>坏盘</b>
ZC20LCWT	0.01	好盘
ZC20LCGY	0.16	好盘
ZC20LCWT	0.01	好盘

如表 5-4 是某次预测结果展示，其中具体预测结果如图所示，第一列为磁盘序列号，第二列为磁盘预测的概率，第三列为预测当前磁盘是否会坏的参考结果。最后模型对这 12 块磁盘预测结束后，共有 3 块磁盘 ZC20LC37、C20KKZA 与 ZC20LCGY 模型认为可能会发生磁盘故障的概率较高，其他的 9 块磁盘预测发送故障的概率较低，从而认定它们是好盘。

## 5.3.2 巡检功能测试

在磁盘巡检模块部署前，需要对磁盘巡检功能进行测试，证明磁盘巡检模块能够正常地识别磁盘坏块。下为磁盘巡检模块的功能测试流程，其中 Hdparm<sup>[77]</sup>为磁盘

工具，可用于查看磁盘的运行相关信息，可修改磁盘相关参数来完成测速、优化等功能，本次将使用 Hdparm 产生磁盘坏块。磁盘巡检模块的功能测试流程如下，相关操作如图 5-3 与图 5-4 所示：

- (1) 对测试服务器进行巡检模块的安装；
- (2) 随机选取该服务器存储磁盘某扇区，用磁盘巡检工具校验；
- (3) 使用 Hdparm 破坏存储磁盘该扇区；
- (4) 使用巡检工具巡检此扇区，发现该扇区已损坏；

```
Use of --make-bad-sector is VERY DANGEROUS.
You are trying to deliberately corrupt a low-level sector on the media.
This is a BAD idea, and can easily result in total data loss.
Please supply the --yes-i-know-what-i-am-doing flag if you really want this.
Program aborted.
[root@TENCENT64 /usr/local/hcbs/scrub/etc]# hdparm --yes-i-know-what-i-am-doing --make-bad-sector 393273400 /dev/sdf
/dev/sdf:
Corrupting sector 393273400 (WRITE_UNC_EXT as pseudo): succeeded
[root@TENCENT64 /usr/local/hcbs/scrub/etc]# dd if=/dev/sdf of=/ox bs=512 skip=393273400 count=1
^Cdd: error reading '/dev/sdf': Input/output error
0+0 records in
0+0 records out
0 bytes (0 B) copied, 46.8522 s, 0.0 kB/s
dd: closing input file '/dev/sdf': Bad file descriptor
```

图 5-3 破坏磁盘扇区

```
[root@TENCENT64 /usr/local/cbs/scrub/etc]# cat scrub_error.conf
[scrub sector detection]
Device Model:   MTFDDAK1T9TBY
Serial Number:  174719D4CB8F
Diskdev:        /dev/sdf
Sector Lba:     393273400
Findtime:       2019/03/06 15:31:09
```

图 5-4 磁盘巡检记录

如图 5-3 所示，将磁盘“/dev/sdf”的 393273400 扇区进行破坏，磁盘将不能进行读写操作，dd 命令将无法执行。如图 5-4 所磁盘巡检记录所示，巡检程序将磁盘“/dev/sdf”的 393273400 扇区错误检测成功，并且形成告警文件。

## 5.4 评价指标

在本单元介绍磁盘故障预测系统的性能时，将作为两部分进行介绍，分别是磁盘故障预测系统相关通用性能的评估指标，和磁盘故障预测应用于巡检后，对磁盘可靠性的提升的评估指标，然后分别对两个子系统的性能测试。上面在设计系统时已经提到，本系统设计为二级系统，整个系统的流程是在磁盘故障预测完成后，将被预测为坏的磁盘再进行磁盘巡检调整。所以首先会预先选取对磁盘故障预测子系

统进行测试，然后对磁盘巡检系统进行评估。

#### 5.4.1 性能评估指标

在对系统的性能进行评估时，所用的指标为业界的通用标准，或者的其他已有参考文献中提出的评估指标。本文有关于其计算方法将会做统一介绍，包括其代表的各项参数意义和性能数值的计算方式。

磁盘故障预测是一个二分类问题，好盘为负样本 Negative，坏盘为正样本 Postive，所以对于按照好、坏盘的预测成功与否，一共有四种预测结果，分别如表 5-5 混淆矩阵所示。

表 5-5 混淆矩阵

混淆矩阵		预测结果		总计
		坏盘	好盘	
真实结果	坏盘	TP	FN	P
	好盘	FP	TN	N
总计		PP	PN	PP + PN = P+N

其中各项数值标识为在一次预测中各项的数值分布，TN 为 Ture Negative 标识真实好盘预测为好盘的数量，TP 为 Ture Postive 表示真实坏盘预测为坏盘数量，FP 为 False Positive 标识真实好盘预测为坏盘的数量，FN 为 False Negative 标识真实坏盘预测为好盘的数量。

在磁盘故障预测问题中，最关心的两个指标为召回率 *Recall* 和误报率 *Fpr*。召回率表示系统能够将多少比例的故障磁盘预测出来，是衡量磁盘故障预测系统的主要评估指标，计算方式如(5-1)所示。

$$Recall = \frac{TP}{TP + FN} \quad (5-1)$$

如式(5-1)所示，其中 TP 为真实坏盘预测为坏盘的数目，FN 为真实坏盘预测为好盘的数目。误报率表示系统错误预警率，计算方式如(5-2)：

$$Fpr = \frac{FP}{FP + TN} \quad (5-2)$$

在误报率中，FP 为将真实好盘预测为坏盘的数目，TN 为将真实好盘预测为好



盘的数目。

### 5.4.2 巡检系统评价指标

在设计磁盘巡检系统时，首要考虑的就是磁盘巡检对业务访问 I/O 的影响，尽量设置对磁盘影响较小的巡检速率。所以本小节将巡检对磁盘的影响进行量化，并且假设磁盘巡检的速率与巡检对磁盘的影响为线性关系，磁盘巡检越高，则对磁盘的影响越大。

在本文的巡检模块中，使用普通巡检来代替一般的巡检方式，指不依据磁盘预测结果的好坏进行巡检速率的调整；使用修正巡检来代替我们的实验方案，即若磁盘预测为坏盘，则将其下发更高速度的巡检配置。

在此我们类似研究<sup>[78]</sup>的方法将巡检对磁盘的影响进行定量分析。其中关键的参数如表 5-6 所示：

表 5-6 巡检系统关键参数

符号标识	符号意义
$\omega$	两次巡检的时间间隔
$\lambda$	普通巡检的坏块发现速率
$\mathcal{X}$	修正巡检对坏块的加速倍数
$MTTD_{normal}$	普通巡检的平均坏块检测时间
$MTTD_{adaptive}$	修正巡检的平均坏块检测时间
$M_{factor}$	$MTTD$ 提升的倍数
$Cost_{normal}$	普通巡检的损失
$Cost_{adaptive}$	修正巡检的损失
$C_{factor}$	修正巡检损失的收益率

如表 5-6 所示共有两列，其中左侧为各符合标识，右侧为符号对应的含义。 $\omega$  为同一扇区的两次巡检的时间间隔，一般为磁盘的巡检周期，此参数是一个固定量。 $\lambda$  为磁盘坏扇区被巡检到的速率，在巡检周期内假设磁盘是均匀巡检的，所以磁盘的坏块将被均匀发现，所以 $\lambda$ 的值将与两次巡检的时间间隔呈反比，记为 $\lambda = 1/\omega$ ，而坏块的平均发现时间与坏块被发现的速率成反比，所以普通巡检的坏块发现时间记为 $MTTD_{normal}$ ，其计算方法如式(5-3)：



$$MTTD_{normal} = \frac{1}{2\lambda} \quad (5-3)$$

在每周期磁盘预测结束后，修正巡检方案将对坏盘进行巡检加速，在此我们设巡检速率的加速系数为  $\mathcal{X}$ ，即修正巡检的坏块发现速率为  $\mathcal{X} * \lambda$ 。在计算修正巡检的 MTTD 时，根据预测的结果分类，需要考虑两部分坏盘情况。对于原本的坏盘预测为坏盘记为 TP，原本的坏盘预测为好盘记为 FN，因为两者的巡检速率不同而 MTTD 计算有所不同，具体  $MTTD_{adaptive}$  的计算如式(5-4)：

$$\begin{aligned} MTTD_{adaptive} &= \frac{TP * \frac{1}{\mathcal{X} * 2\lambda}}{TP + FN} + \frac{FN * \frac{1}{2\lambda}}{TP + FN} \\ &= \frac{1}{\mathcal{X} * 2\lambda} * Recall + \frac{1}{2\lambda} * (1 - Recall) \end{aligned} \quad (5-4)$$

其中  $Recall$  是磁盘预测的召回率，表示正确预测的坏盘占有所有应该被预测到的比例。到此为止，修正巡检的评价故障探测时间仅和  $Recall$  有关，平均故障探测时间的提升  $M_{factor}$  与修正巡检和普通巡检的故障探测时间有关，具体为两者的比值，其具体计算如式(5-5)：

$$\begin{aligned} M_{factor} &= \frac{MTTD_{factor}}{TPMTTD_{normal}} \\ &= \frac{1}{\mathcal{X}} * Recall + (1 - Recall) \end{aligned} \quad (5-5)$$

由式(5-5)可知，修正巡检相对于普通巡检效率的提升，仅与预测的召回率和巡检加速比例有关，与其他参数没有关系。

巡检过程对磁盘的损失与巡检持续的时间，和巡检的速率有关。假设巡检持续的时间为  $T$ ，在计算普通巡检对磁盘的损失时，因为对每一种预测结果都是一样的巡检速率，所以那么在此时间内普通巡检对磁盘的损失记如式(5-6)：

$$Cost_{fixed} = T * \lambda * (P + N) \quad (5-6)$$

其中  $P$  为坏盘的集合，记为  $P = TP + FN$ ； $N$  为好盘的集合，记为  $N = FP + TN$ 。对于修正巡检来说，其对磁盘的损失根据预测的结果分类，需要考虑两部分坏盘情况。对于预测为坏盘的集合  $PP$  使用加速巡检，记  $PP = TP + FP$ ；预测为好盘的集合  $PN$  使用常速巡检，记  $PN = FN + TN$ 。所以修正巡检的损失  $Cost_{adaptive}$  和相对与普通巡

检的提升 $C_{factor}$ 分别如式(5-7)与式(5-8):

$$Cost_{adaptive} = T * \lambda * (\mathcal{X} * PP + PN) \quad (5-7)$$

$$\begin{aligned} C_{factor} &= \frac{Cost_{adaptive} - Cost_{fixed}}{Cost_{fixed}} \\ &= \frac{\mathcal{X} * PP + PN - (P + N)}{(P + N)} \\ &= \frac{(\mathcal{X} - 1) * PP}{(P + N)} \end{aligned} \quad (5-8)$$

其中  $PP=TP+FP$ ，记为所有被预测的坏盘； $P+N$  为所有的磁盘总数。在后续的性能测试部分，我们将对不同预测结果情况下，对应的巡检速率变化时，巡检的性能提升情况。巡检的性能由 $M_{factor}$ 巡检坏块检测时间的变化率，和巡检对前台的影响优化因子 $C_{factor}$ 体现。

## 5.5 性能测试

本小结为磁盘故障预测系统性能评估测试，将分两部分进行描述：第一部分是磁盘故障预测系统的性能测试，主要是关于磁盘故障预测的召回率和误报率性能参数。第二部分是磁盘故障预测的结果应用与巡检系统，分析巡检系统性能的提升情况。

### 5.5.1 预测系统性能测试

上一小结已经介绍了用于磁盘故障预测系统的相关实验参数以及性能指标，本小结将完成基于所选择的测试数据完成性能评估，并且给出系统性能测试相关结果。

在本章的评估测试中，我们将使用 Company-T 和 Backblaze 的各三种型号磁盘作为整体数据源对算法进行测试。同时采用第三章中特征处理进行特征筛选和构建，最后得出共 19 维属性。在本节的性能测试中，分别包括对算法模型的横向和纵向比较，其中横向比较为本文提出的基于时序随机森林和其他算法的比较，纵向比较为验证时序数据分配算法和时序模型更新算法的有效性。

为了横向评估决策树算法的实用性，我们评估了 CART 算法，BP 算法，LR 算

法，以及 SVM 算法的性能结果。对于其他模型，在我们的数据集上训练、调参后，选择最好的实验性能结果进行比较，实验结果如图 5-5 所示。

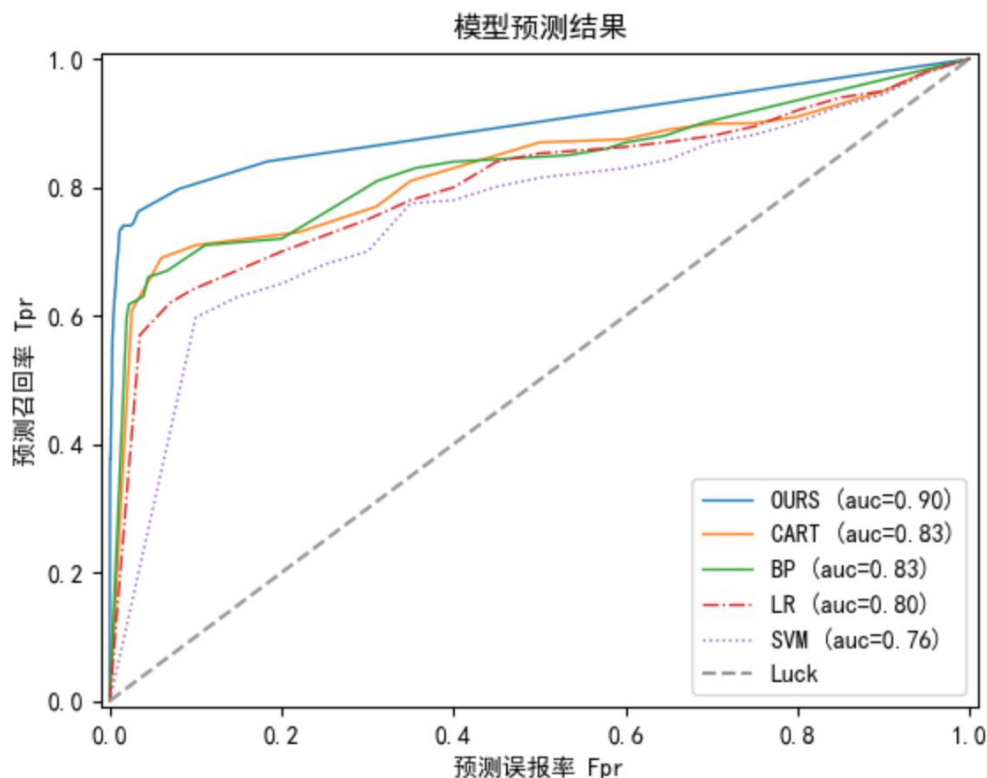


图 5-5 多种不同算法的 ROC 曲线

如图 5-5 所示，蓝色曲线为本文算法，灰色的线为随机预测曲线，ROC 曲线是一种比较分类算法性能的工具，其横轴是误报率，纵轴是召回率，两者围成的面积 AUC 反映算法的预测效果。本文提出的基于时序随机森林算法与传统的机器学习算法相比有明显的优势，在误报率都为 1% 时，其召回率约为 78%，ROC 为 0.9 相比最好的 CART 算法高出为 10%。在磁盘故障预测模型的准确率上，本文的算法优于其他机器学习算法。

在纵向比较中，为了验证本文提出的两种算法有效性，将对其分别进行测试，比较是否选用对应的算法从而验证其性能提升。在实验过程中，使用 Company-T 和 Backblaze 的各三种型号磁盘作为整体数据源对算法进行测试，其中变量为使用时序数据处理算法，和使用时序模型优化算法，具体的实验设计如下，实验结果如表 5-7：

实验 1：随机森林不使用时序数据处理算法，不使用时序模型优化算法

实验 2：随机森林仅使用时序数据处理算法

实验 3：预测模型仅使用时序模型优化算法

实验 4：随机森林使用时序数据处理算法，且使用时序模型优化算法

表 5-7 横向对比算法有效性

实验编号	Company-T 数据		backblaze 数据	
	FPR	FAR	FPR	FAR
实验 1	69.84%	1.96%	71.51%	1.06%
实验 2	77.62%	1.93%	80.49%	0.97%
实验 3	70.63%	0.94%	72.26%	0.49%
实验 4	78.28%	0.98%	81.94%	0.92%

如表 5-7 所示为各配置下的预测结果。其中每一行为对应的配置信息，用于比较本文提出算法对模型优化与否，纵轴是两种数据集的预测情况。分析两种数据集结果，由表中实验 1 和 2 可知，在使用时序数据处理算法后，在误报率相同的基础上，召回率约提升 11.13%，证明算法可有效提高预测召回率；由表中实验 1、3 可知，在使用时序模型优化算法后，在召回率不变的基础上，误报率下降约 52.0%，证明时序模型优化算法可有效降低预测模型的误报率；由表中实验 2、3、4 可知，在将两者算法结合使用后，召回率与误报率的优化效果皆有稳定提升。

### 5.5.2 故障预测与磁盘巡检

磁盘故障预测应用与巡检时，将预测为坏盘的巡检速度提高，从而更快的发现坏块以此来降低 MTTF，但同样会对增加磁盘的压力从而造成负面影响。所以本小节将会分析不同预测效果下，对巡检加速系数更改的收益与损失。

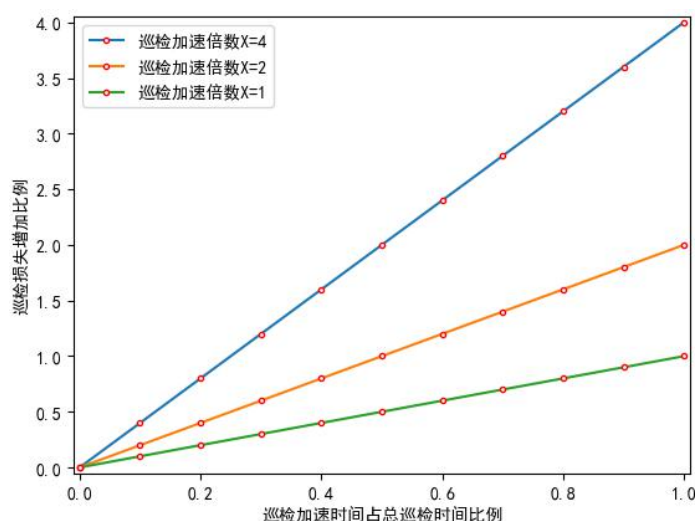


图 5-6 加速巡检时间占比与对磁盘损失关系

上图为巡检在不同的巡检系数下，对磁盘的损失 $C_{factor}$ 图像。我们根据式(5-7)可知，巡检的时间越长对磁盘的影响越大。且巡检加速系数 $\mathcal{X}$ 越大对磁盘的影响越大。如图5-6所示，横轴是加速巡检时间占所有巡检时间的比例，纵轴是加速巡检状态下，对磁盘的损失比例。蓝色直线为巡检加速系数 $\mathcal{X}$ 为4时对磁盘的增加的损失，橙色直线为巡检加速系数 $\mathcal{X}$ 为2时对磁盘的增加的损失，绿色直线为常速巡检时的损失，在比较时以常速巡检对磁盘的损失为基准。

除此之外我们希望了解在不同的磁盘故障预测准确率下，加速巡检时间占比与磁盘MTTD提升比例的对比，所以需要首先求出加速巡检的部分，如上小节的分析，加速巡检占比为预测为坏盘的数量比上全部磁盘的数量；而通过MTTD提升的比例进而求出 $M_{factor}$ ，所以我们可以根据不同预测结果进行作图，如图5-7所示。

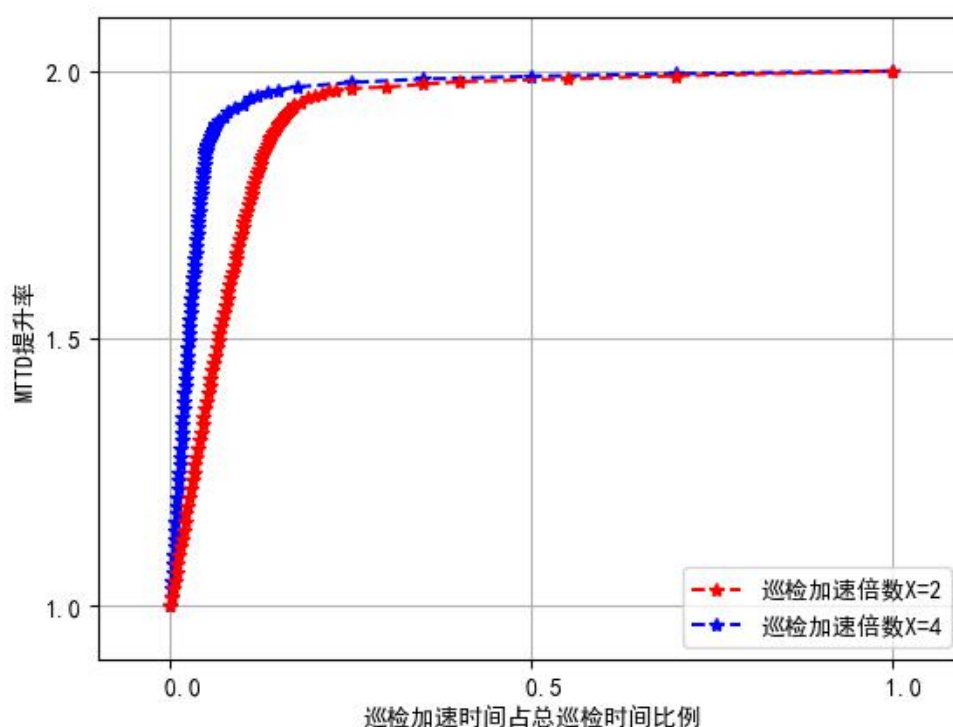


图 5-7 巡检加速时间占比与 MTTD 收益曲线

如图 5-7 所示，是预测程序在不同的准确率与召回率下，在加速系数  $\alpha$  不同时对坏盘 MTTD 的影响。其中横轴是巡检在加速状态下持续的时间，纵轴是 MTTD 提升的比例，不同的曲线代表加速系数的不同。随着巡检加速时间比例的增大，坏盘 MTTD 的收益也不断增高，同时可知在磁盘加速时间很短时 MTTD 的提升已经非常明显，且加速系数越大，将对 MTTD 的收益增加越快。当巡检加速参数设为 4 时，在加速巡检时间为 5.0% 时，MTTD 的收益为 152.6%。

随着磁盘处于加速时间的增加，MTTD 的收益会增加，但与此同时磁盘的 COST 也在不断增加，下面将比较不同预测结果时不同的准确率和召回率时，巡检 MTTD 的收益与磁盘 COST 的关系，实验结果如图 5-8 所示。

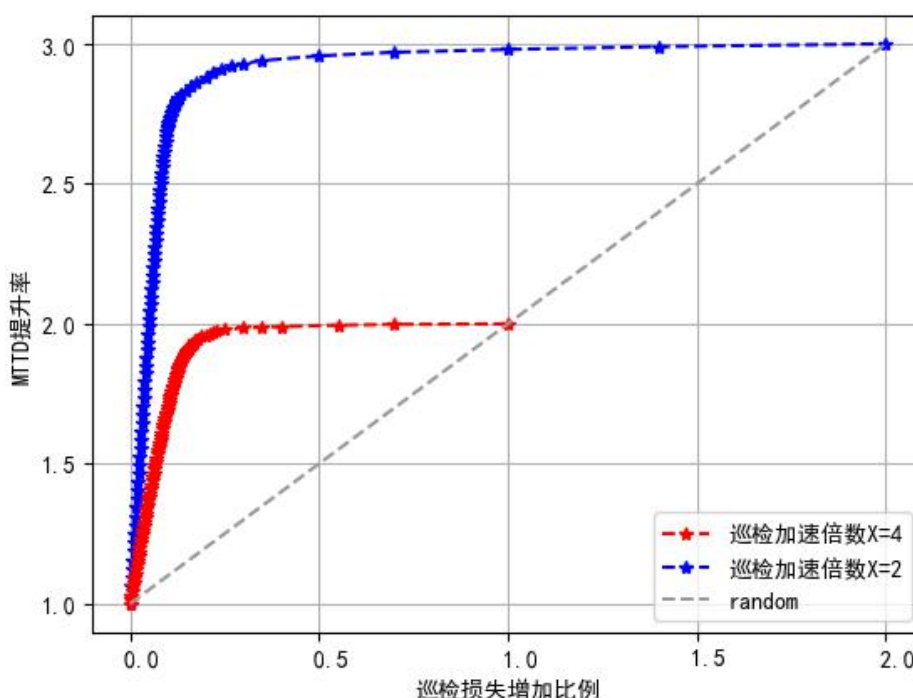


图 5-8 巡检加速时间占比与 MTTR 收益曲线

图 5-8 展示了加速巡检的收益与损失的关系，图中不同线代表不同的巡检加速系数  $X$ ，线上每一个点则代表了在不同预测召回率-误报率时，对应的巡检 MTTR 的收益与磁盘 COST 的关系。其中横轴的磁盘 COST，纵轴是 MTTR 的收益。图中蓝线代表加速倍数为 4 的图像，在对磁盘无减速巡检时，磁盘的 MTTR 为基准值 1.0；在对磁盘预测结果不同时，对应的 MTTR 提升和磁盘 COST 也不同，图中 random 线为随机预测器在磁盘故障预测时对磁盘影响，在对磁盘随机进行加速时，得到的 MTTR 提高与磁盘 COST 为正比例关系。蓝线为在加速系数为 4 时，可知磁盘的 COST 增加 4.8% 时，磁盘的 MTTR 的提升达 217.3%。

在实际生产环境中，可结合每周期的预测准确率情况，对巡检加速系数进行动态配置，可在对磁盘相对较小的 COST 下得到非常大的 MTTR 提升收益。

## 5.6 本章小结

本章对磁盘故障预测系统，磁盘巡检系统进行了详细地测试，并给出了相关的评价指标和测试结果。在数据集方面，使用了 Company-T 和 Backblaze 各三种型号磁盘作为整体进行实验，使用时序数据模型后，在同样的误报率下召回率提高 11.13%；



使用优化的时序模型后，在同样的召回率下误报率降低 52.0%。另外使用多种经典分类算法与我们进行比较，证明本文提出的时序随机森林模型质量优于经典算法。

在磁盘故障预测与巡检应用结合时，根据实际巡测结果画出了磁盘损失与 MTDD 收益曲线，加速巡检时间占比为 5.0% 时，平均故障探测时间降低了 152.6%；磁盘的磁盘巡检负载增加 4.8% 时，平均故障探测时间降低了 217.3%，可大幅度减少磁盘巡检开销，降低平均故障探测时间。该实验结果可以用于指导实际生产系统，通过时刻预测准确率和误报率，动态调整巡检加速配置从而得到最大的 MTDD 收益和较小的磁盘 COST，证明了磁盘故障预测技术对磁盘巡检系统的有效性。

## 6 总结与展望

### 6.1 本文工作总结

回顾本文，本文站在宏观的角度上看待磁盘故障预测问题，并总结了磁盘故障预测的主要方法，对比了现有方法的预测性能以及弊端。针对现有预测模型的一些主要弊端，设计了基于时序数据的随机森林预测模型，提出了时序数据分配算法与时序模型更新算法对模型进行改进。设计并实现了磁盘数据采集系统、磁盘故障预测系统、磁盘巡检系统以及中心监控系统，并且在大型数据中心部署持续监控磁盘情况，保障数据安全。

基于时序数据的随机森林基于传统的随机森林模型拓展，增加了时序数据功能，使得随机森林可以接受时序信息。本文提出的时序数据分配算法支持根据数据的时序特征，对决策树进行动态设置权值从而提高模型的准确率。通过随机森林的各决策树分配不同的时间样本数据，从而实现决策树到时间序列的权值映射，赋予随机森林的时序性。本文提出的时序模型更新算法负责在模型更新时，对随机森林中各决策树进行权重的更新。算法学习历史预测样本数据，对预测正确的决策树进行权值增加，使模型不断学习到最新的坏盘特征，实现模型的更新。

本文结合实际工业生产环境，为数据中心设计并实现了一套智能化磁盘故障预测应用原型系统。原型系统包括预测数据的采集模型，故障预测模块，磁盘巡检模块以及中心监控模块。它们彼此作为有机整体协同合作，共同实现磁盘故障预测，提升存储系统可靠性。数据采集模块为故障预测模块提供数据采集、保存服务，为日后的故障分析提供数据保障。故障预测系统对磁盘进行持续异常监控，对可能存在故障的磁盘故障预测，保障磁盘的可靠性。磁盘巡检系统对磁盘周期性巡检，发现并修复磁盘潜在的故障，从而降低实际读写的错误概率。中心监控模块负责提供对各个模块的监控和告警功能，保障各个模块的正常工作运行，对接与运维人员和各项告警系统的接入。

## 6.2 未来研究工作展望

回顾本文关于磁盘故障预测的研究，可以从以下三个方面开展未来的研究工作，分别是磁盘运行数据积累方面，为数据中心积累数据价值；磁盘故障预测模型方面，不断提高预测召回率降低误报率；以及将磁盘故障预测与工业结合，结合数据中心中其他系统，通过减低磁盘的故障率来提高其他系统的可靠性。

在磁盘故障预测等故障诊断问题中，训练数据质量决定了预测结果的上限。磨刀不误砍柴工，我们可以在故障预测之前，系统性地将此问题相关的数据部分进行梳理归纳，从而得到训练时更有用的信息。在以往的磁盘故障预测问题中，大多只采用的磁盘的 S.M.A.R.T 属性作为特征，但是由于厂家实现的不同导致其数据含义可能存在差异。在后续的研究中可考虑在应用层面上收集磁盘相关信息，如通过收集系统日志中磁盘坏块数据对磁盘坏盘进行预测；或是通过见过系统 I/O 负载，并且将数据实时记录，开展磁盘 I/O 负载对磁盘故障的影响的研究等。对于磁盘业务方面的数据都可以进行记录，并且用于磁盘故障预测系统，通过数据源维度的增加从而提升预测模型的准确率。

在预测模型方面，现有磁盘故障预测模型对时序信息考虑较少，后续可以从模型角度进一步进一步深入研究。磁盘故障预测是一个分类问题，目前现有的很多机器学习方法可对其进行尝试解决。但模型预测时最重要的预测的召回率和误报率，传统的机器学习算法不能有效地体现出磁盘数据的变化特征，从而忽视了该问题最重要的变化属性。在后续的研究中，在模型方面可设计深度学习网络，在网络中加入属性变化的时间信号，从而可以对磁盘进行时间维度上建模，提高预测准确率。如可基于 LSTM 网络进行磁盘故障预测，它可将历史的磁盘数据信息在网络中存储，使得可以区分磁盘时序变化信息，从而在最快的时间发现磁盘属性的变化对坏盘模型进行学习。

考虑磁盘故障预测问题与巡检或者其他应用结合，单纯的磁盘故障预测能够解决的问题十分有限，在工业系统中对预测的故障磁盘进行下架处理时，对预测的准确率要求很高，目前的数据质量或者是模型能力暂未达到工业要求。但是在与存储可靠性结合的其他领域，可以借用磁盘故障预测对其性能进行优化，而不需要很高的预测精度就能提升很高的效率。如磁盘巡检问题，在召回率和误报率效果随着时

间周期不稳定变化时，通过动态设置巡检参数，可以保证巡检的提升在一个相对稳定的范围中。之后的研究方向可以使磁盘故障预测与相似领域进行结合，使用磁盘故障预测对实际问题进行优化，从而提升存储系统的可靠性。比如可以使磁盘故障预测与存储系统三副本备份结合，若三副本数据有两部分处于预测故障的磁盘时，开启数据备份程序将三副本数据增加至四副本以上，避免在磁盘故障后造成系统处于只读状态或导致数据丢失。另外也可考虑对系统数据进行分级管理，对于可靠性要求更高的数据，如果其数据落在潜在故障率较高的磁盘上时，将数据进行备份或者迁移，从而保证数据的可靠性。

磁盘故障预测系统作为数据中心中数据安全模块，担起存储可靠性的最后一道防线。它在故障发生之前对系统做出预警，提醒运维人员提前进行数据备份等操作。有了磁盘故障预测系统，存储系统的可靠性将迈上一个新的台阶。

## 致 谢

时光荏苒，白驹过隙，转眼间研究生生涯已接近尾声。不由得想起了本科毕业前来实验室青涩的自己，满怀着对科研的憧憬与家人的期盼，跨进了国家光电实验室的大门，充实的研究生旅程就此开启篇章。

感谢实验室的指导老师们。李春花副教授作为我研究生的导师，治学严谨的工作态度与孜孜不倦的科研追求让我印象深刻；逐字逐句修改我第一篇英语论文时的一丝不苟和谆谆教诲，让我感触颇深。周可教授深厚的学术造诣和渊博的专业知识让我深深敬佩，也是我努力和奋斗的榜样，像一座灯塔照亮我前行的航线。王桦副教授的温文尔雅和对科研的精益求精的热情让我深受触动，程海燕秘书的认真负责和对同学们的热情让我记忆犹新。

感谢华中科技大学遇到的朋友们。感谢指导我的师兄师姐们，每个迷茫苦闷的时刻，是你们指引我正确的方向。感谢实验室的同学们，每个刺骨寒风的清晨去上课的日子，是你们陪着我一起坚持；每个炎炎夏日的深夜刷题面试的时光，是你们陪着我一起努力。感谢任课老师们，每个风霜雨雪都激情澎湃授课的时光，是你们让我懂得了什么是情怀。感谢国家光电实验室的工作人员，无论春夏秋冬都站岗到深夜，默默守护着我们的安全。

感谢在腾讯实习期间一起工作的同事们。在近两年的实习经历中，感谢你们指点和教导我，让我成长为一名合格的程序员；感谢你们相信和支持我，让我独立开发的程序运行在数十万的计算机集群中；感谢你们包容和鼓励我，让我有信心和能力面对未来一切的挑战。感谢实习期间遇到的渣渣团们，让我在深圳这座陌生的城市感受到更多的温暖，那些日子有了你们才更加的精彩。

感谢我的家人们。在离家武汉上学已有七年时间，感谢你们信任和支持我，让我在陌生的环境，不惧任何困难和挑战；感谢你们背后鼓励和安慰我，让我在灰暗无助的日子里，心中亮起爱的星光；感谢你们资助我的学业，让我在坎坷的求学道路上，心里一直有所依靠。感谢我的女朋友王倩，在我困难无助的时候温暖我受伤的心，在我得意的时候一起分享快乐的喜悦，在我迷茫绝望的时候在我身旁陪伴鼓励。请你们放心，现在的我已经成长为独立自强、有所担当的人，面对未来将自信昂扬。

## 参考文献

- [1] 金海, 袁平鹏, 石柯. 网格计算.第二版. 北京: 电子工业出版社, 2004.
- [2] Mesnier M, Ganger G R, Riedel E. Object-based storage. IEEE Communications Magazine, 2003, 41(8): 84-90.
- [3] Prabhakaran V, Bairavasundaram L N, Agrawal N, et al. IRON file systems. SIGOPS Oper. Syst. Rev, 2005, 39(5): 206-220.
- [4] Ghemawat S, Gobioff H, Leung S. The Google file system. SIGOPS Oper. Syst. Rev, 2003, 37(5): 29-43.
- [5] 毛波. 盘阵列的数据布局技术研究[博士学位论文]. 武汉: 华中科技大学图书馆, 2010.
- [6] 罗东健. 大规模存储系统高可靠性关键技术研究[博士学位论文]. 武汉: 华中科技大学图书馆, 2011.
- [7] 吴素贞. 磁盘阵列高可用技术研究[博士学位论文]. 武汉: 华中科技大学图书馆, 2010.
- [8] 刘军平. 磁盘存储系统可靠性技术研究[博士学位论文]. 武汉: 华中科技大学图书馆, 2011.
- [9] Vishwanath K V, Nagappan N. Characterizing cloud computing hardware reliability. in: Proceedings of the 1st ACM symposium on Cloud computing. Indianapolis, Indiana, USA: ACM, 2010. 193-204
- [10] Schroeder B, Gibson G A. Disk failures in the real world: what does an MTTF of 1,000,000 hours mean to you. in: Proceedings of the 5th USENIX conference on File and Storage Technologies. San Jose, CA: USENIX, 2007. Article No. 1
- [11] Shvachko K, Kuang H, Radia S, et al. The Hadoop Distributed File System. in: Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies. Washington, DC, USA: IEEE Computer Society, 2010. 1-10
- [12] Rizzo L. Effective erasure codes for reliable computer communication protocols. in: ACM SIGCOMM Computer Communication Review. New York, NY, USA: ACM,



1997. 24-36.
- [13] 李明强. 磁盘阵列的纠删码技术研究[硕士学位论文]. 北京: 清华大学图书馆, 2011.
- [14] 朱云锋. 分布式存储系统中基于纠删码的容错技术研究[硕士学位论文]. 合肥: 中国科学技术大学图书馆, 2014.
- [15] S.M.A.R.T. [EB/OL]. [2019/3/20]. <https://zh.wikipedia.org/zh-hans/S.M.A.R.T>.
- [16] Yang Y, Li W, Zhang M, et al. Disk failure prediction model for storage systems based on disk SMART technology. in: International Journal of Computers and Applications. 2015. 111-119.
- [17] 杨寅. 存储系统可靠性关键技术研究[博士学位论文]. 武汉: 华中科技大学图书馆, 2013.
- [18] 胡维. 基于智能预警和自修复的高可靠磁盘阵列关键技术研究[硕士学位论文]. 武汉: 国防科学技术大学图书馆, 2010.
- [19] 董勇, 蒋艳凰, 卢宇彤, 等. 面向磁盘故障预测的机器学习方法比较. 计算机工程与科学, 2015,37(12):2200-2207.
- [20] Vilalta R. Disk failure prediction in heterogeneous environments. in: International Symposium on Performance Evaluation of Computer and Telecommunication Systems. Seattle, WA, USA: IEEE, 2017
- [21] Bairavasundaram L N, Arpaci-Dusseau A C, Arpaci-Dusseau R H, et al. An analysis of data corruption in the storage stack. in: ACM Transactions on Storage. New York, NY, USA: ACM, 2008. 1-28.
- [22] Bairavasundaram L N, Goodson G R, Pasupathy S, et al. An analysis of latent sector errors in disk drives. in: SIGMETRICS Perform. San Diego, California, USA: ACM, 2007. 289-300.
- [23] Pinheiro E, Weber W, Andr L, et al. Failure trends in a large disk drive population. in: Proceedings of the 5th USENIX conference on File and Storage Technologies, San Jose, CA: USENIX Association, 2007.
- [24] Hamerly G, Elkan C. Bayesian approaches to failure prediction for disk drives. in: Proceedings of the Eighteenth International Conference on Machine Learning. San

- Francisco, CA, USA: Morgan Kaufmann, 2001. 202-209.
- [25] Hughes G F, Murray J F, Kreutz-Delgado K, et al. Improved disk-drive failure warnings. in: IEEE Transactions on Reliability, 2002,51(3):350-357.
- [26] Murray J F, Hughes G F, Kreutz-Delgado K. Hard drive failure prediction using non-parametric statistical methods. in: ICANN, 2003.
- [27] Murray J F, Hughes G F, Kreutz-Delgado K. Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application. in: International Conference on Robotics and Automation. San Diego, CA, USA: Mach. Learn. Res, 2005. 783-816.
- [28] 张超. 高性能磁盘阵列自修复技术研究[博士学位论文]. 长沙: 国防科学技术大学图书馆, 2008.
- [29] Li J, Stones R J, Wang G, et al. Hard drive failure prediction using Decision Trees. in: 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. Atlanta, GA, USA: IEEE, 2017. 164:55-65.
- [30] Mahdisoltani F, Stefanovici I, Schroeder B. Proactive error prediction to improve storage system reliability. in: Proceedings of the 2017 USENIX Conference on Usenix Annual Technical Conference. Santa Clara, CA: USENIX Association, 2017, 391-402.
- [31] Monitoring Hard Disks with SMART | Linux Journal[EB/OL]. [2019/3/20]. <https://www.linuxjournal.com/article/6983>.
- [32] Hong B, Madhyastha T. The relevance of long-range dependence in disk traffic and implications for trace synthesis. in: 13th NASA Goddard Conference on Mass Storage Systems and Technologies. Monterey, CA, USA, USA: IEEE, 2005.
- [33] Eckart B, Chen X, He X, et al. Failure Prediction Models for Proactive Fault Tolerance within Storage Systems. in: IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems. Baltimore, MD, USA: IEEE. 2008(02).
- [34] Nurmanova V, Bagheri M, Zollanvari A, et al. A New Transformer FRA Measurement Technique to Reach Smart Interpretation for Inter-disk Faults. in: IEEE

- Transactions on Power Delivery, 2019(04).
- [35] Ferreira V C, Nery A S. A smart disk for in-situ face recognition. in: IEEE International Parallel and Distributed Processing Symposium Workshops. Vancouver, BC, Canada: IEEE. 2018(06).
- [36] He W, Du D H. Smart: An approach to shingled magnetic recording translation. in: Proceedings of the 15th Usenix Conference on File and Storage Technologies. Berkeley, CA, USA: USENIX, 2017(02). 121-133.
- [37] 宋云华. 基于 S.M.A.R.T.预测故障磁盘的研究[硕士学位论文]. 南京: 南京大学图书馆, 2014.
- [38] 柳永康. 基于机器学习的磁盘故障预测系统[硕士学位论文]. 武汉: 华中科技大学图书馆, 2015.
- [39] 兰云龙. 基于深度学习的磁盘健康度评估与故障预测技术研究[硕士学位论文]. 杭州: 杭州电子科技大学图书馆, 2017.
- [40] 廖玉兰. 基于分布式的决策树方法研究[硕士学位论文]. 成都: 电子科技大学图书馆, 2011.
- [41] Ryszard S. Michalski, Igor Mozetic, Nada Lavrac, et al. The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. in: Proceedings of the Fifth AAAI National Conference on Artificial Intelligence. Philadelphia, Pennsylvania: AAAI Press, 1986. 1041-1045.
- [42] Wenger E. Communities of practice and social learning systems: the career of a concept. in: Social learning systems and communities of practice, 2010. 179-198.
- [43] Quinlan J R. Induction of Decision Trees. Mach. Learn. Machine Learning, 1986, 1(1):81-106.
- [44] Tax D M J, Duin R P W. Data domain description using support vectors. ESANN'1999 proceedings - European Symposium on Artificial Neural Networks Bruges, 1999, 251-256.
- [45] Vapnik V N. The nature of statistical learning theory. Springer-Verlag, 1995.
- [46] Quinlan J R. Improved use of continuous attributes in C4. 5. Journal of artificial intelligence research, 1996,4:77-90.

- [47] Lyden D, Young A Z, Zagzag D, et al. Id1 and Id3 are required for neurogenesis, angiogenesis and vascularization of tumour xenografts. *Nature*, 1999,401(6754):670.
- [48] Spits H, Couwenberg F, Bakker A Q, et al. Id2 and Id3 inhibit development of CD34+ stem cells into predendritic cell (pre-DC) 2 but not into pre-DC1: evidence for a lymphoid origin of pre-DC2. *Journal of Experimental Medicine*, 2000, 192(12): 1775-1784.
- [49] 杨霖, 周军, 梅红岩, 等. ID3 改进算法研究. *软件导刊*, 2017, 16(08): 21-24.
- [50] 朱琳, 杨杨. ID3 算法的优化. *软件*, 2016, 37(12): 89-92.
- [51] 王永梅, 胡学钢. 决策树中 ID3 算法的研究. *安徽大学学报(自然科学版)*, 2011, 35(03): 71-75.
- [52] Quinlan J R, et al. Bagging, boosting, and C4.5. in: *AAAI'96 Proceedings of the thirteenth national conference on Artificial intelligence*, 1996.
- [53] 陈杰, 邬春学. 决策树 C4.5 算法改进与应用. *软件导刊*, 2018, 17(10): 88-92.
- [54] 刘佳, 王新伟. 一种改进的 C4.5 算法及实验分析. *计算机应用与软件*, 2008, 25(12): 260-262.
- [55] Lawrence R L, Wright A. Rule-based classification systems using classification and regression tree (CART) analysis. *Photogrammetric engineering and remote sensing*, 2001, 67(10): 1137-1142.
- [56] Lewis R J. An introduction to classification and regression tree (CART) analysis. in: *Annual Meeting of the Society for Academic Emergency Medicine*. San Francisco, California: Department of Emergency Medicine, 2000.
- [57] Liaw A, Wiener M, Others. Classification and regression by randomForest. 2002, 2(3): 18-22.
- [58] De Andres S A. Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 2006, 7(1): 3.
- [59] Pal M. Random forest classifier for remote sensing classification. *International Journal of Remote Sensing*, 2005, 26(1): 217-222.
- [60] Ho T. Random decision forests, 1995.09.
- [61] Ho T K. The Random Subspace Method for Constructing Decision Forests. *IEEE*

- Trans. Pattern Anal. Mach. Intell., 1998, 20(8): 832-844.
- [62] 刘波. 基于 SVDD 特征选择的随机森林高分辨率遥感影像变化检测[硕士学位论文]. 兰州: 兰州交通大学图书馆, 2018.
- [63] Pearson K C. On lines and planes of closest fit to systems of points in space. Philosophical Magazine, 1901, 2(6): 559-572.
- [64] 苏雅茹. 高维数据的维数约简算法研究[博士学位论文]. 合肥: 中国科学技术大学图书馆, 2012.
- [65] Schwarz T J E, Miller E L, Long D D E, et al. Disk scrubbing in large archival storage systems. in: Proceedings of the The IEEE Computer Society's 12th Annual International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems. Washington, DC, USA: IEEE, 2004. 409-418.
- [66] 吴庆民. 大数据环境下数据容错技术研究[博士学位论文]. 合肥: 中国科学院大学图书馆. 2016.
- [67] Elerath J G. RAID-6 system reliability dependence on recovery, disk scrubbing, and group size. in: 2016 Annual Reliability and Maintainability Symposium, 2016. Tucson, AZ, USA: IEEE.
- [68] Li N, Xie Q, Wang Y, et al. A page lifetime-aware scrubbing scheme for improving reliability of Flash-based SSD. IEICE Electronics Express, 2017, 14(22): 201-224.
- [69] sg\_verify - invoke SCSI VERIFY command(s) on a block device - Linux Man Pages (8). 2019/3/20. [https://www.systutorials.com/docs/linux/man/8-sg\\_verify/](https://www.systutorials.com/docs/linux/man/8-sg_verify/).
- [70] Iliadis I, Haas R, Hu X, et al. Disk Scrubbing Versus Intra-disk Redundancy for High-reliability Raid Storage Systems. 2008, 36(1): 241-252.
- [71] The Apache Software Foundation. 2019/3/22. <http://www.apache.org/>.
- [72] Nginx news[EB/OL]. 2019/3/22. <http://nginx.org/>.
- [73] MapReduce-腾讯云. 2019/3/22. <https://cloud.tencent.com/product/emr>.
- [74] scikit-learn: machine learning in Python — scikit-learn 0.20.3 documentation. 2019/3/22. <https://scikit-learn.org/stable/>.
- [75] 云服务器-腾讯云. 2019/3/22. <https://cloud.tencent.com/product/cvm>.
- [76] The Best Unlimited Online Backup and Cloud Storage Services. 2019/3/22.

<https://www.backblaze.com/>.

[77] hdparm - ArchWiki. <https://wiki.archlinux.org/index.php/hdparm>.

[78] Jiang T, Zhou K. Scrub Unleveling: Achieving High Data Reliability at Low Scrubbing Cost. in: 2019 Design, Automation & Test in Europe Conference & Exhibition, 2019. Florence, Italy : IEEE .



## 附录 1 攻读学位期间参与的科研项目

1. 国家重点研发计划“网络空间安全”重点专项子项“基于新型密码算法的大数据保护机制与方法”，2016YFB0800402，2016.7-2021.6
2. 华中科技大学-腾讯智能联合实验室校企合作“存储缓存优化和磁盘故障预测研究”项目，经费编号 0231187706，2016.7-2019.6

## 附录 2 攻读学位期间取得的科研成果

### 1. 已发表的论文

Li C, **Xie W**, Zhou K. Efficient Binary-Encoding Access Control Policy Combination for Large-Scale Collaborative Scenarios: 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (TrustCom), 2018. p. 560-566.

### 2. 已申请的专利

李春花, 周可, **谢伟睿**. 一种访问控制策略合成方法[p]. 中国, 国家发明专利, 专利公开号: 201810797439X. 2018-07-19.