

采用数据降维的固态硬盘故障检测方法

王宇菲, 董小社, 王龙翔, 陈维多, 陈衡

(西安交通大学电子与信息学部, 710049, 西安)

摘要: 针对固态硬盘 SMART 数据包含大量高维数据特征导致故障检测准确率不理想的问题, 结合固态硬盘 SMART 数据的时序特点, 基于门控循环单元对传统自动编码器的结构做了改进, 提出了一种基于门控循环单元稀疏自动编码器降维的固态硬盘故障检测方法(GAL)。首先利用固态硬盘 SMART 数据训练 GRUAE 模型, 然后利用 GRUAE 模型中的编码器作为降维的工具, 对固态硬盘的原始高维 SMART 数据进行降维, 减少固态硬盘原始 SAMRT 数据中噪声特征的影响并突出与数据特点更加相关的特征, 以提高故障检测的准确率, 最后基于降维过的低维 SMART 数据利用长短时记忆网络进行故障检测。选取阿里巴巴固态硬盘数据集, 对多种人工智能算法在准确率、召回率和 $F_{0.5}$ 方面进行了比较。实验结果表明: 相比于没有采用任何降维手段, 采用 GAL 方法可使两种闪存类型的固态硬盘的故障检测准确率、召回率和 $F_{0.5}$ 分别提高 4%、5%、4% 和 4%、8%、5%, 分别达到 97%、95%、97% 和 97%、96%、97%; GAL 方法的故障检测准确率、召回率和 $F_{0.5}$ 分别超出 WEFR 对比方法 53%、25%、43%。

关键词: 固态硬盘; 故障检测; 降维; 稀疏自动编码器; 门控循环单元

中图分类号: TP319 **文献标志码:** A

DOI: 10.7652/xjtuxb202211018 **文章编号:** 0253-987X(2022)11-0176-10

A Solid-State Drive Fault Detection Method with Dimensionality Reduction

WANG Yufei, DONG Xiaoshe, WANG Longxiang, CHEN Weiduo, CHEN Heng

(Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: The massive high-dimensional SMART data attributes tend to cause a low precision in the detection of solid-state drive (SSD) faults. For this reason, this paper proposes an SSD fault detection method-GAL based on dimensionality reduction with GRU sparse auto-encoder based on an improved structure of the auto-encoder with gated recurrent unit (GRU) according to temporal characteristics of SSD SMART data. First, the GRUAE model is trained with SSD SMART data. Then the encoder of GRUAE model is used as the dimensionality reduction tool to reduce the original high-dimensional SSD SMART data. This is to reduce the influence of noise features in original SSD SAMRT data and highlight the features more relevant to data characteristics to improve the precision of fault detection. Finally, long short-term memory is used for fault detection with low-dimensional SSD SMART data. The precision, recall and $F_{0.5}$ score of various algorithms are compared based on the dataset from Alibaba. The experimental results show that GAL improves the fault detection precision, recall and $F_{0.5}$ -score of the two types of flash SSDs by 4%, 5%, 4% and 4%, 8% and 5%, respectively, reaching 97%, 95%, 97% and 97%,

96%, 97%, respectively, compared with the condition with no dimensionality reduction. In addition, the precision, recall and $F_{0.5}$ score of GAL are respectively 53%, 25% and 43% higher than those of the method compared

Keywords: fault detection; dimensionality reduction; sparse auto-encoder; solid state drive; gated recurrent unit

近年来,固态硬盘在诸如阿里巴巴、谷歌、微软等大型公司里的普及程度越来越高^[1],但是固态硬盘往往只有有限的使用寿命,随着磨损老化最终失效是在所难免的^[2-6]。故障检测技术往往能够提前发现问题,保证大规模存储系统的可靠性和可用性,降低数据丢失的风险。不少已有工作针对机械硬盘的故障检测技术进行了研究^[4,7-10]。文献[4]开发了一种基于成本敏感排序的机器学习模型,根据硬盘在将来的错误倾向对硬盘故障进行排序分类。Zhang 等根据硬盘局部扇区的错误构建了故障检测模型^[8]。Xiao 等基于在线随机森林提出了硬盘故障预测模型^[9]。Wang 等基于长短时记忆网络改进了生成对抗网络的结构,通过生成虚拟硬盘故障数据以扩充样本集规模,提高了小样本情况下硬盘故障检测准确率^[10]。

然而,由于固态硬盘复杂的特性,只有少数工作基于非公开的数据研究了固态硬盘的故障检测。文献[3]基于随机森林方法研究了不同自我监测分析与报告技术(SMART)特征对固态硬盘故障检测的重要程度。谷歌针对其内部固态硬盘分析了数据特点并构建了故障检测模型^[11-12]。Sarkar 等基于设备固件功能提供的特征研究了固态硬盘的故障检测技术^[13]。

新型固态硬盘通常包含大量的高维 SMART 数据,这会给众多人工智能算法的效果带来负面影响^[14]。降维能够减少原始数据中噪声特征的影响,并突出与数据特点更加相关的特征,提升算法效果^[15]。国内外关于传统降维方法的研究主要有主成分分析(PCA)^[16]、线性判别分析^[17]、奇异值分解(SVD)^[18]、遗传算法^[19]和因子分析(FA)^[20]等。文献[21]利用堆叠自动编码器(AE)对旋转电机的振动谱信号进行降维,提高了旋转电机故障检测准确率。文献[22]基于堆叠自动编码器对农作物监控数据降维,提升了分类精度。文献[23]结合了自动编码器和基于集成思想的分类器提出了一种新的数据降维方法。文献[24]利用自动编码器的降维特性解决了高维地震反演问题。不过这些降维方法多用于处理线性数据,降维后的低维数据通常为原始高维

数据的线性组合,在处理高维非线性数据时存在不小的局限性。并且已有工作普遍基于固态硬盘的原始高维 SMART 数据直接进行分析并结合人工智能算法构建故障检测模型,在模型构建的数据预处理阶段有不小的局限性,存在可优化的空间。

为解决上述问题,本文结合固态硬盘 SMART 数据的时序特点,基于门控循环单元(GRU)对自动编码器的结构做了改进,提出了一种采用 GRU 稀疏自动编码器(GRUAE)降维的固态硬盘故障检测方法(GAL)。AE 能够在无监督方式下学习数据编码,可以非线性地提取数据中最重要的特征,对于大量高维非线性数据降维效果较好^[25]。GAL 首先利用固态硬盘数据训练 GRUAE 模型,在 GRUAE 模型训练完成后利用其中的编码器作为降维的工具,对固态硬盘的原始高维 SMART 数据进行降维,减少固态硬盘原始 SMART 数据中噪声特征的影响并突出与数据特点更加相关的特征,随后基于降维过的低维 SMART 数据利用长短时记忆网络(LSTM)进行故障检测。实验结果表明,相比于没有采用任何降维手段,GAL 使两种闪存类型的固态硬盘故障检测准确率、召回率和 $F_{0.5}$ 分别提高了 4%、5%、4%和 4%、8%、5%,分别达到 97%、95%、97%和 97%、96%、97%。GAL 的故障检测准确率、召回率和 $F_{0.5}$ 分别超出对比方法 53%、25%、43%。

1 背景知识

1.1 门控循环单元

GRU 通过调节单元内信息流的门控结构可以捕获不同时间尺度的依赖关系,且不需要单独的记忆单元^[26]。

GRU 在 t 时刻的激活函数 h_t^i 计算公式如下

$$h_t^i = (1 - z_t^i)h_{t-1}^i + z_t^i\hat{h}_t^i \tag{1}$$

式中: z_t^i 为更新门; $\hat{h}_t^i$ 为候选激活函数。

更新门决定了一个单元在 t 时刻更新多少内容,计算公式如下

$$z_t^i = \sigma(\mathbf{W}x_t + \mathbf{U}h_{t-1})^j \tag{2}$$

式中: x 为输入数据; $\mathbf{W}$ 和 $\mathbf{U}$ 为权重矩阵; σ 为

Sigmoid函数。候选激活函数 $\hat{h}_t^j$ 的计算与传统循环单元类似,计算公式如下

$$\hat{h}_t^j = \tanh(\mathbf{W}x_t + \mathbf{U}(r_t \odot h_{t-1}))^j \quad (3)$$

式中: $\odot$ 是基于元素的乘法; r_t 是一组重置门。重置门计算公式如下

$$r_t^j = \sigma(\mathbf{W}x_t + \mathbf{U}h_{t-1})^j \quad (4)$$

1.2 自动编码器

AE 是一种学习模型,旨在通过无监督学习从数据中提取表示特征。AE 的结构包括编码器和解码器两部分,如图 1 所示。编码器试图从输入数据中提取潜在代码,将输入数据映射到低维特征空间;解码器试图从潜在代码中重构出尽可能接近原始输入数据的数据片段。

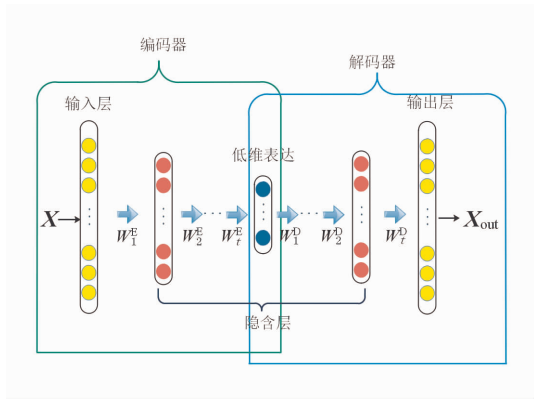


图 1 自动编码器结构示意图

Fig. 1 Architecture of auto-encoder

编码器和解码器的计算公式如下

$$\mathbf{h} = \sigma_1(\mathbf{W}^E \mathbf{X} + \mathbf{b}) \quad (5)$$

$$\mathbf{Y} = \sigma_2(\mathbf{W}^D \mathbf{h} + \mathbf{d}) \quad (6)$$

式中 $\mathbf{X}=[x_1, \cdots, x_n]$ 表示输入数据; n 为该输入数据的维度; σ_1 和 σ_2 为激活函数; $\mathbf{h}$ 为隐含层特征向量; $\mathbf{W}^E$ 和 $\mathbf{W}^D$ 分别为编码器与解码器的权重参数矩阵,其序列长度与网络层数相关。 $\mathbf{Y}$ 为重构后的向量。AE 的目标是使重构误差最小化,计算式如下

$$L_{AE} = \frac{1}{n} \sum_{i=1}^n \| \mathbf{y}_i - \mathbf{x}_i \|^2 \quad (7)$$

2 GAL 方法

在真实环境中,固态硬盘的故障是随着使用时间的推移而逐渐出现的。因此,固态硬盘可靠性特性具有较强的时间相关性。相较于传统线性网络,GRU 更善于捕捉样本的时序特点,提取与时间相关的特征。采用 GRU 作为 AE 模型的编码器来拟合固态硬盘样本的概率分布函数,使编码器可以更好

地学习到固态硬盘 SMART 数据的时序特性,提取其中的潜在代码,将原始输入数据映射到低维特征空间。同时解码器也采用 GRU 以便更好地捕捉数据的时序特性,将数据从低维空间重构出原始状态。对于整个 AE 网络,其输入样本维度与输出结果维度相同,训练网络时的标签就是输入样本自身,即整个网络输入等于输出。当训练结束后,得到的最终结果就是编码器与解码器各自的权重参数矩阵序列,此时再将原始样本依次与编码器的权重参数矩阵序列相乘可得到该数据的低维表示,从而实现高维数据的压缩降维,然后该数据的低维表示即能作为其他处理的输入。GAL 方法如图 2 所示,主要包含 GRUAE 的训练、降维和故障检测 3 个部分,分为 3 个主要步骤。

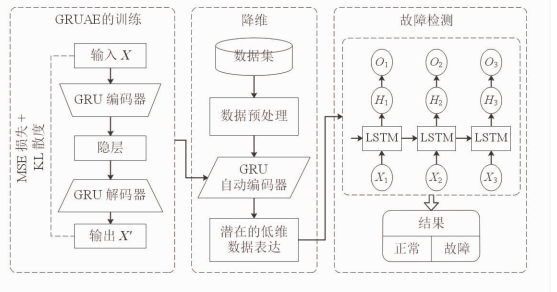


图 2 GAL 方法结构图

Fig. 2 Structure of GAL method

(1)GRUAE 的训练。对于输入的原始数据,训练用于降维的 GRUAE 模型,通过 GRU-编码器对原始输入数据进行编码,在隐含层将输入映射到低维特征空间,如此在保留了原始输入数据特点的基础上实现了数据维度的削减;为确保低维特征空间里的数据编码包含了原始输入数据的特点以及各数据特征之间的联系,随后通过 GRU-解码器对隐含层输出解码,将数据从低维状态重构成原始高维状态。在每一轮的训练过程中,采用 MSE 均方误差+KL 散度作为度量方式,通过稀疏惩罚项抑制隐含层神经元的输出,计算该重构输出与原始输入之间的误差,使网络具有稀疏的特性。这样即使隐含层神经元数很多,GRUAE 模型仍然可以学习到输入数据的重要特征。

(2)降维。在 GRUAE 模型训练完成后,取出其中的 GRU-编码器作为降维的工具,对固态硬盘数据集的原始 SMART 数据进行正则化、归一化等预处理,随后利用 GRU-编码器对数据进行降维,将固态硬盘 SMART 数据映射到一个低维特征空间,以减少原始 SMART 数据中噪声特征的影响并

突出与数据特点更加相关的特征,形成一个新的低维数据集。

(3)故障检测。由于固态硬盘 SMART 数据时序特性较强,而 LSTM 善于捕获样本的时序特性和提取时间相关特征。鉴于此,基于降维过的低维数据集,采用 LSTM 进行故障检测,输出诊断结果。

2.1 GRUAE 模型

GRUAE 模型包含两个独立的 GRU 层,结构如图 3 所示。对于样本数为 k 的 d 维样本集 $\mathbf{x} \in \{\mathbf{x}_k^d\}$,GRU-编码器提取输入数据潜在的特征,将输入映射成一个 d' 维的低维向量 $\mathbf{h} \in \{h_k^{d'}\}$;GRU-解码器基于 $\mathbf{h} \in \{h_k^{d'}\}$ 低维特征空间尽可能的重构出接近 d 维原始输入数据的 $\mathbf{z} \in \{z_k^d\}$,随后 GRU-编码器会用于生成后续故障检测用的低维数据集。

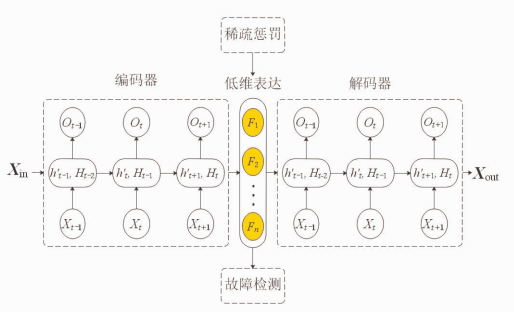


图 3 GRUAE 模型结构图

Fig. 3 Structure of GRUAE model

对于模型的结构,如果隐含层包含的节点多于输入层,则会影响 AE 学习特征的能力。因此,在隐含层节点中加入了稀疏惩罚项作为约束,通过抑制隐含层神经元的输出,使网络达到了稀疏的效果。这样即使隐含层神经元数多,AE 仍然可以学习到输入数据的重要特征。

编码器过程表达式如下

$$\mathbf{h} = f_{\theta}(\mathbf{x}) = A^E(\mathbf{W}\mathbf{x} + \mathbf{b}) \quad (8)$$

编码器的参数集表示为: $\theta = \{\mathbf{W}, \mathbf{b}\}$; $\mathbf{W}$ 为权重矩阵; $\mathbf{b}$ 为偏移向量; A^E 是编码器的激活函数。相似地,重构输出表示为

$$\mathbf{z} = g_{\theta'}(\mathbf{h}) = A^D(\mathbf{W}'\mathbf{h} + \mathbf{b}') \quad (9)$$

解码器的参数集表示为: $\theta' = \{\mathbf{W}', \mathbf{b}'\}$; $\mathbf{W}'$ 为权重矩阵; $\mathbf{b}'$ 为偏移向量; A^D 是解码器的激活函数。因此,损失函数定义为

$$J = \min \frac{1}{k} \sum_{i=1}^k \|\mathbf{z}_i - \mathbf{x}_i\|^2 \quad (10)$$

假设 $a_j(x)$ 表示隐含层中第 j 单元的激活量,则第 j 单元的平均激活量为

$$\hat{p}_j = \frac{1}{k} \sum_{i=1}^k a_j(x(i)) \quad (11)$$

式中: k 表示样本数。 p 表示稀疏常数,为了确保大多数神经元处于不活跃状态,设 $\hat{p}_j$ 等于一个接近于 0 的值,然后在损失函数中加入一个稀疏惩罚项来惩罚 $\hat{p}_j$ 。选择 KL 散度作为惩罚项 R 的表达式

$$R = \sum_{j=1}^{s_2} \text{KL}(p \| \hat{p}_j) \quad (12)$$

式中: s_2 为隐含层单元数。 $\text{KL}(p \| \hat{p}_j)$ 为 KL 散度,具体表达式如下

$$\text{KL}(p \| \hat{p}_j) = p \ln \frac{p}{\hat{p}_j} + (1-p) \ln \frac{1-p}{1-\hat{p}_j} \quad (13)$$

添加了稀疏惩罚项的 GRUAE 模型的优化目标函数定义如下

$$J(\mathbf{W}, \mathbf{b}) = \frac{1}{k} \sum_{i=1}^k \|\mathbf{x}(i) - \hat{\mathbf{x}}(i)\|^2 + \frac{\lambda}{2} \sum_{f=1}^{n_l-1} \sum_{i=1}^{S_f} \sum_{j=1}^{S_{f+1}} (\mathbf{W}_{ij}(f))^2 \quad (14)$$

式中: λ 为权重衰减系数; n_l 为神经网络层数; S_f 为第 f 层的神经元数; S_{f+1} 为第 $f+1$ 层的神经元数; $\mathbf{W}_{ij}(f)$ 表示第 f 层神经元 i 和第 $f+1$ 层神经元 j 之间连接关系所关联的参数。在模型训练的反向传播阶段,需要计算当前输入数据和输出数据之间的误差,以及除输出层之外的每一层网络里神经元的误差以更新网络参数 $(\mathbf{W}, \mathbf{b})$ 。因此,包含了稀疏惩罚表达式的损失函数表示为

$$J_{\text{sparse}}(\mathbf{W}, \mathbf{b}) = J(\mathbf{W}, \mathbf{b}) + \beta R \quad (15)$$

式中: β 为稀疏惩罚系数,本文实验设置为 3。

对于输入的原始高维固态硬盘 SMART 数据,初始化网络的参数,在每一次迭代中,前向传播过程首先计算出隐含层输出结果,然后根据隐含层结果计算出输出层结果;反向传播过程通过 Adam 方法更新网络参数,以最小化添加了稀疏惩罚项的损失函数,反复循环直至收敛。GRUAE 模型训练过程的伪代码如下。

输入 数据样本; Epoch; Batch

输出 GRUAE 模型

- 1 初始化网络权重 $\mathbf{W}$
- 2 初始化网络偏置 $\mathbf{b}$
- 3 for $e=1$ to Epoch do
- 4 for batch_number=1 to Batch do
- 5 计算重构约束矩阵
- 6 计算局部约束矩阵
- 7 for $k=1$ to K do

```
8    前向传播过程
9    计算目标约束矩阵
10   反向传播过程,优化目标
11   更新网络参数
12 if loss 不再下降 then
13   停止训练,返回结果
14 end if
```

2.2 降维

GRUAE 模型训练完成后,取出其中的编码器,对于样本数为 k 的 d 维样本集 $\mathbf{x} \in \{x_k^d\}$,利用编码器提取输入数据潜在的特征,将输入映射成一个 d' 维的低维样本集 $\mathbf{h} \in \{h_k^{d'}\}$,计算过程如式(8)所示。

2.3 LSTM 模型

LSTM 通过定义和维护单元状态来调节信息流以获取长期的时间依赖性。存储单元状态 C_{t-1} 与中间输出 h_{t-1} 和随后的输入 x_t 相互作用,以确定内部状态向量的哪些元素应该根据前一时间步的输出和当前时间步的输入进行更新、维护或消失。LSTM 的计算公式如下

$$i_t = \sigma(x_t \mathbf{U}_i + h_{t-1} \mathbf{W}_i) \tag{16}$$

$$f_t = \sigma(x_t \mathbf{U}_f + h_{t-1} \mathbf{W}_f) \tag{17}$$

$$o_t = \sigma(x_t \mathbf{U}_o + h_{t-1} \mathbf{W}_o) \tag{18}$$

$$\hat{C}_t = \tanh(x_t \mathbf{U}_g + h_{t-1} \mathbf{W}_g) \tag{19}$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \tag{20}$$

$$h_t = \tanh(C_t) * o_t \tag{21}$$

式中:操作符 $*$ 表示对应元素逐个相乘; σ 表示 Sigmoid 激活函数; i_t 、 f_t 和 o_t 分别表示当前输入数据在 t 时刻的输入门、遗忘门和输出门状态; $\hat{C}_t$ 是一个候选隐藏状态,由当前输入和前一个隐藏状态计算得出; $\mathbf{W}_i$ 、 $\mathbf{W}_f$ 、 $\mathbf{W}_o$ 和 $\mathbf{W}_g$ 分别表示训练过程中需要学习的权重矩阵; $\mathbf{U}_i$ 、 $\mathbf{U}_f$ 、 $\mathbf{U}_o$ 和 $\mathbf{U}_g$ 分别为对应单元的系数矩阵; C_t 为单元内部的记忆; h_t 表示记忆单元的最终输出。LSTM 单元通过各种门控结构,可以捕获数据在短时间序列和长时间序列内部的复杂特征。

3 数据集概览与预处理

3.1 数据集概览

本文实验选用了来自阿里巴巴的固态硬盘数据集^[27]。该固态硬盘数据集包含了固态硬盘的 SMART 数据和诸如时间戳、设备序列号等基本信息。需要注意的是不同厂家提供的 SMART 参数可能会不一样,并且某些参数会因为设备类型不同而代表不同的含义。因此,提取了数据集中数据记

录较完整并且 Flash 类型为 MLC 和 3D TLC 的两个型号的固态硬盘在 2019 年全年内的数据进行实验测试。两种固态硬盘的型号分别为 MA2 和 MC1,数据的基本信息见表 1,其采样间隔均为 1 天。

表 1 MA2、MC1 数据集的基础参数

Table 1 Basic parameters of dataset MA2 and MC1

数据集参数	数值	
	MA2	MC1
容量/GB	800	1 920
时间跨度/a	1	1
健康盘数	84 076	163 338
故障盘数	452	8 546
正样本数	30 687 985	59 618 418
负样本数	45 692	1 855 058
处理后数据集大小/GB	13	27

3.2 参数信息和预处理

数据集中的 SMART 数据包含 Raw 值(Raw)和 Normalized 值(Norm)两种赋值种类,其中的 Norm 是根据厂家非公开的自定义公式由 Raw 计算而来。因为有些 Norm 可能会损失一些数据精度,而其对应的 Raw 对于固态硬盘的健康变化可能更为敏感,所以同时采用了 Raw(R)和 Norm(N)作为实验参数,MA2 和 MC1 两种型号的固态硬盘包含的 SMART 参数信息及其赋值类型见表 2。

不同 SMART 参数的单位不一样,参数值所跨的范围差别也很大,为了保证不同参数权重的公平性,采用参数尺度缩放对数据进行归一化,公式如下

$$x_n = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{22}$$

式中: x 是参数的原始值; $x_{\max}$ 和 $x_{\min}$ 分别为该参数的最大值和最小值。

4 实验结果

实验使用了准确率、召回率和 $F_{0.5}$ 3 个指标来衡量不同机器学习算法检测故障的效果。从实践经验来看,一旦一个固态硬盘被判定为故障,不管判定正确或错误,管理员将停止使用该固态硬盘以进行进一步检查。鉴于将一个健康的固态硬盘判定为故障所造成的代价远远大于将一个可能故障的固态硬盘错误地判定为健康。因此,本文实验采用 $F_{0.5}$ 代替 F_1 ,使准确率的重要程度为召回率的两倍。相关定义如下

$$P = \frac{T_p}{T_p + F_p} \tag{23}$$

表 2 MA2、MC1 数据集包含的 SMART 参数名称及其赋值类型

Table 2 The SMART parameters and the corresponding assignment value type contained in MA2/MC1 dataset

SMART 参数名称	赋值类型	
	MA2	MC1
Read Error Rate		N&R
Reallocated Sectors Count	N&R	N&R
Power On Hours	N&R	N&R
Power Cycle Count	N&R	N&R
Available Reserved Space	N&R	N&R
SSD Program Fail Count	N&R	N&R
SSD Erase Fail	N&R	N&R
SSD Wear Leveling Count		N&R
Unexpected Power Loss	N&R	N&R
Unused Reserved Block Count		N&R
SATA Downshift Error Count	N&R	N&R
End-to-End error	N&R	N&R
Uncorrectable Errors	N&R	N&R
Command Timeout		N&R
Airflow Temperature	N&R	
Unsafe Shutdown Count	N&R	
Temperature Celsius	N&R	N&R
Hardware ECC Recovered		N&R
Reallocation Event Count		N&R
Current Pending Sector Count	N&R	N&R
Uncorrectable Sector Count		N&R
UltraDMA CRC Error Count	N&R	
Media Wearout Indicator	N&R	
Total LBAs Written	N&R	
Total LBAs Read	N&R	
Power Loss Protection Failure	N&R	

$$R = \frac{T_P}{T_P + F_N} \tag{24}$$

$$F_{0.5} = \frac{(1 + 0.5^2)PR}{0.5^2P + R} \tag{25}$$

式中： T_P 为真阳性； F_P 为假阳性； F_N 为假阴性。

实验将数据集中的数据分别按照 7:3 的比例随机划分为训练集与测试集进行测试。GRUAE 模型的编码器为具有 1 个隐含层的 GRU,输入层的单元数与对应型号固态硬盘 SMART 数据的维度相等,隐含层有 10 个单元,时序长度为 3,输出层单元数为 10,dropout 设置为 0.2;解码器为具有 1 个隐含层的 GRU,输入层单元数为 10,隐含层单元数为

10,输出层有线性映射并采用 tanh 作为激活函数,输出层的单元数与对应型号固态硬盘 SMART 数据的维度相等,dropout 设置为 0.2,学习率为 0.001。

LSTM 的结构包含输入层、两层 LSTM 隐含层以及输出层,输出层有线性映射,采用 Sigmoid 作为激活函数。输入层的单元数为降维后的固态硬盘 SMART 数据维度,隐含层单元数为 100,输出层单元数为 1,dropout 设置为 0.2,学习率为 0.001。

GRUAE 模型中编码器的输出层单元数为 10,即为降维后的数据维度大小,在确定该数据维度大小时,利用 PCA 方法,设定主成分占投影特征的方差比例为 99%,即包含的主成分对原变量能够解释 99%,自动计算得出该数据维度大小。

训练过程中的优化器为 Adam。图 4 给出了训练过程中 GRUAE 模型在 MA2 和 MC1 两个固态硬盘数据集上损失的变化过程。可以观察到,经过一段时间的训练,模型的损失已经下降至接近 0 并稳定下来,表明模型的训练已经收敛。

实验针对 MA2 和 MC1 两个型号的固态硬盘数据,分别训练了各自的 GRUAE 模型,随后基于 GRUAE 模型对固态硬盘 SMART 数据进行降维操作。基于降维后的数据,对比了多层感知器(MLP)、随机森林(RF)、逻辑回归(LR)、决策树(DT)、支持向量机(SVM)、GRU 和本文方法 7 种人工智能算法的故障检测的效果,并且与文献[23]的方法进行了比较。

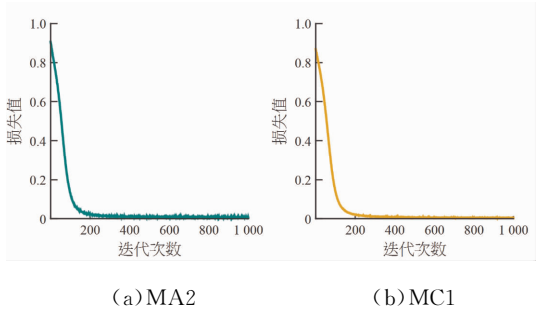


图 4 GRUAE 模型在 MA2、MC1 两个数据集上训练时损失的变化情况

Fig. 4 Loss changes of GRUAE model during training on datasets

实验所用平台为一台高性能服务器,具体配置如下:两个 Intel Xeon E5-2620 v4 @ 2.10 GHz 处理器、94G DDR4 内存、4TB 存储空间。在软件方面,服务器安装了 Ubuntu 18.04.5 LTS 操作系统,其内核版本为 Linux version 4.15.0-171-generic,

编译器版本为 gcc version 9.3.0。方法实现基于 PyTorch 1.7.1 和 scikit-learn 0.24.1。

4.1 MA2 数据集结果

MA2 的准确率如图 5 所示,在没有采用任何降维手段的情况下,GRU 和 LSTM 两个在处理时序数据方面具有优势的深度学习算法要明显好于其他 5 种传统机器学习算法。在进行降维后,对于所有算法来说,准确率基本都有或多或少的提升。其中 MLP 和 DT 的提升幅度较大,平均接近 20%,其他方法的提升相对 MLP 和 DT 来说稍低一些,RF 和 SVM 的提升幅度分别为 6% 和 13%。对于 LR,降维带来的提升效果不太明显,平均提升效果不到 3%。虽然 GRU 与 LSTM 的基础效果已经比较出色了,但是降维依旧可以带来 3%~4% 的提升。并且由于 LSTM 本身的网络结构比 GRU 复杂,因此效果也稍微好一点。本文所提方法 GAL 即 GRUAE+LSTM 的准确率超过了 97%,是所有方法里最好的一个。

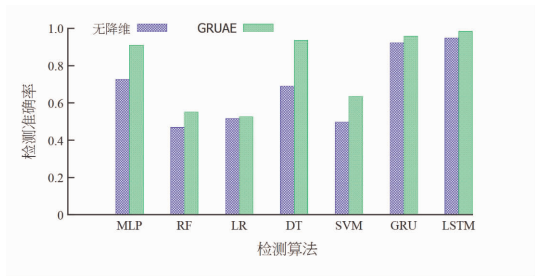


图 5 在 MA2 数据集上的准确率对比
Fig. 5 Comparison of precision on MA2 dataset

MA2 的召回率如图 6 所示,与准确率类似,在没有采用任何降维手段的情况下,GRU 和 LSTM 两个在处理时序数据方面具有优势的深度学习算法要明显好于其他 5 种传统机器学习算法。在进行降维后,有些算法的召回率有很大的提升,例如 DT,平均提高幅度将近 40%;也有一些算法的提升效果比较微小,例如 LR 和 SVM,平均提升幅度大约为 3%。MLP、RF、GRU 以及 LSTM 的提升效果比较接近,平均增幅为 5% 左右。同样地,尽管 GRU 和 LSTM 的基础效果已经比较出色了,但是本文的 GRUAE 降维依旧可以提升其召回率,并且本文所提方法 GAL 的表现依然是所有方法里最好的一个。

MA2 的 $F_{0.5}$ 如图 7 所示,整体趋势与准确率相似,在进行降维后,多种算法的效果基本都有提升,整体来说,MLP、DT 和 SVM 的相对提升幅度较多,平均在 15% 左右。GRU 和 LSTM 的效果非常

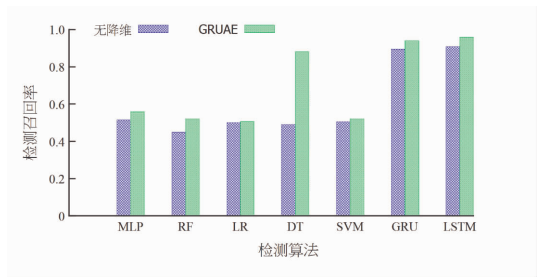


图 6 在 MA2 数据集上的召回率对比
Fig. 6 Comparison of recall on MA2 dataset

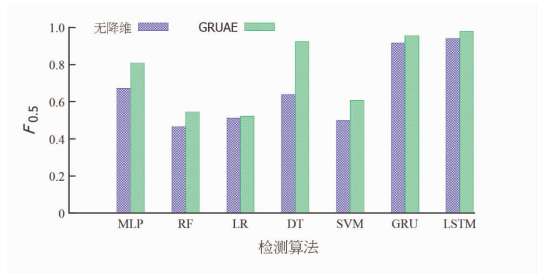


图 7 在 MA2 数据集上的 $F_{0.5}$ 对比
Fig. 7 Comparison of $F_{0.5}$ -score on MA2 dataset

接近,均好于其他算法。由于本文所提方法 GAL 的准确率和召回率在所有方法中均为最好,因此其 $F_{0.5}$ 也是最好的一个。

4.2 MC1 数据集结果

MC1 的准确率如图 8 所示,与 MA2 的准确率类似,GRU 和 LSTM 在处理时序数据方面具有优势的深度学习算法要明显好于其他 5 种传统机器学习算法。相比于没有降维的时候,降维后基本所有算法都有提升,其中 RF 的提升效果尤为明显,平均提高了约 35%。MLP、DT 和 SVM 的提升幅度也比较可观。对于 MLP、DT、SVM、GRU 和 LSTM 来说,准确率分别提升了 6.5%、8.5%、12.5%、4% 和 4%。虽然 GRU 和 LSTM 作为基础效果最好的两个方法,准确率分别为 92% 和 94%,但是经过 GRUAE 的降维其准确率依旧能够得到提升,其中 LSTM 的效果稍好于 GRU。本文所提方法 GAL 作

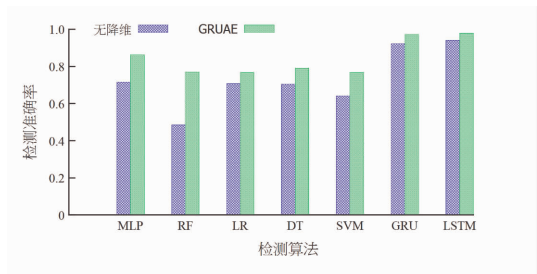


图 8 在 MC1 数据集上的准确率对比
Fig. 8 Comparison of precision on MC1 dataset

为整体效果最好的方法,准确率几乎达到了 98%。

MC1 的召回率如图 9 所示,在不采用任何降维手段的情况下,GRU 和 LSTM 依然是基础效果最好的方法,原始召回率分别达到了 90%和 91%;DT 在 5 种传统机器学习算法中效果最好,原始召回率为 56%。在进行降维后,DT 依然是提升幅度最大的算法,平均提高幅度将近 20%;也有一些算法的提升效果比较微小,例如 LR 和 SVM,幅度不到 3%。经过 GRUAE 降维后 GRU 和 LSTM 的召回率均提高了 5%左右,达到了 95%和 96%,因此本文所提方法 GAL 的效果依旧最好。

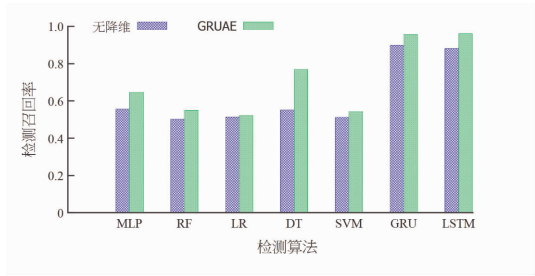


图 9 在 MC1 数据集上的召回率对比
Fig. 9 Comparison of recall on MC1 dataset

MC1 的 $F_{0.5}$ 如图 10 所示,整体趋势与准确率相似,在进行降维后,多种算法的效果基本都有提升,经过 GRUAE 降维后,MLP、DT 和 SVM 的相对提升幅度最多,平均在 9%左右。由于本文所提方法 GAL 的准确率和召回率在所有方法中均为最好,因而其 $F_{0.5}$ 也是最好的一个。

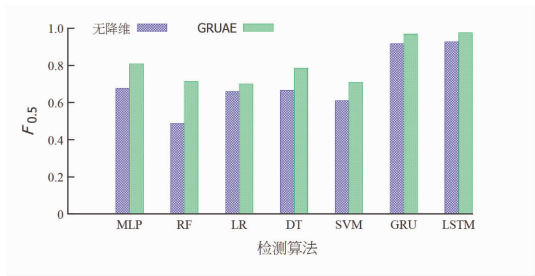


图 10 在 MC1 数据集上检测的 $F_{0.5}$ 对比
Fig. 10 Comparison of $F_{0.5}$ -score on MC1 dataset

4.3 与 WEFR 的对比结果

为进一步评估本文所提方法 GAL 效果,对比了文献[27]所提的 WEFR 方法。因为 GAL 方法基于 GRU 改进了传统 AE 的结构,使 GRUAE 模型的编码器可以更好地学习到固态硬盘 SMART 数据的时序特性,提取其中的潜在代码,减少固态硬盘原始高维 SMART 数据中噪声特征的影响并突出与数据特点更加相关的特征,将原始输入数据映射

到低维特征空间。因此,经过 GRUAE 模型降维处理,多种人工智能算法的故障检测效果得到了较好地提升。

如图 11 和图 12 所示,在 MA2 数据集上 WEFR 的准确率、召回率和 $F_{0.5}$ 分别为 57%、32%和 49%。除了 RF 和 LR,其他方法基于 GRUAE 降维后都达到了更好的效果,效果最好的 LSTM 的准确率、召回率和 $F_{0.5}$ 分别为 97.8%、95%和 97%,分别超过 WEFR 40.8%、63%和 48%。在 MC1 数据集上 WEFR 的准确率、召回率和 $F_{0.5}$ 分别为 49%、18%和 36%。基于 GRUAE 降维后的所有算法的效果均好于 WEFR,效果最好的 LSTM 的准确率、召回率和 $F_{0.5}$ 分别为 97%、96%和 96.8%,分别超过 WEFR 48%、78%和 60.8%。因为固态硬盘的 SMART 数据时序特点较强,而 WEFR 基于随机森林方法,在处理时序数据时存在一定的局限性,所以整体准确率和召回率有不少可提升的空间。

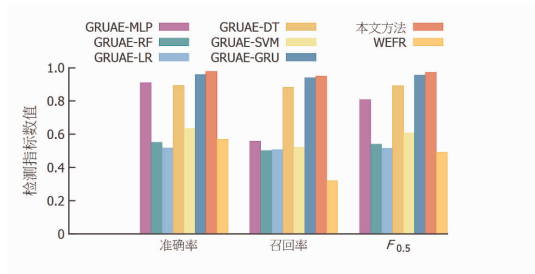


图 11 在 MA2 数据集上本文方法与 WEFR 的准确率、召回率、 $F_{0.5}$ 对比结果

Fig. 11 Comparison of precision/recall/ $F_{0.5}$ -score between the proposed method with WEFR on MA2 dataset

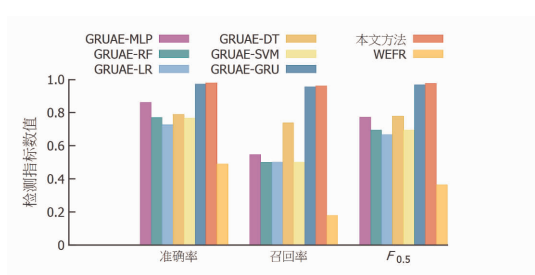


图 12 在 MC1 数据集上本文方法与 WEFR 的准确率、召回率、 $F_{0.5}$ 对比结果

Fig. 12 Comparison of precision/recall/ $F_{0.5}$ -score between the proposed method with WEFR on MC1 dataset

实验结果表明,GRUAE 模型可以学习固态硬盘 SMART 数据的时序特点,对固态硬盘原始高维 SMART 数据进行降维,在保证降维后的数据包含了原始数据特点的情况下,提高了多种人工智能算法的故障检测准确率。在两个型号的固态硬盘数据集上,相比于无降维情况下的固态硬盘故障检测,

GAL使检测的准确率、召回率和 $F_{0.5}$ 分别提高了4%、5%、4%和4%、8%、5%。相较于其他方法, GAL的检测准确率、召回率和 $F_{0.5}$ 超出了53%、25%、43%。并且, GAL在多种评价指标方面均给出了最好的效果, 充分说明了方法的有效性。

5 结 论

为解决固态硬盘 SMART 数据的高维性对传统机器学习算法故障检测效果带来的负面影响, 本文结合 GRU 与 AE 提出了一种基于 GRU 稀疏自动编码器降维的固态硬盘故障检测方法 GAL。首先通过学习固态硬盘 SMART 数据的特点训练 GRUAE 模型, 随后利用 GRUAE 模型中的编码器对高维固态硬盘 SMART 数据进行降维, 提取其中的潜在代码, 减少固态硬盘原始 SMART 数据中噪声特征的影响并突出与数据特点更加相关的特征, 随后基于降维后的低维 SMART 数据利用 LSTM 进行固态硬盘故障检测。实验结果表明, 相比于没有采用任何降维手段, GAL 使两种闪存类型的固态硬盘故障检测准确率、召回率和 $F_{0.5}$ 分别提高了4%、5%、4%和4%、8%、5%, 分别达到97%、95%、97%和97%、96%、97%。GAL 的故障检测准确率、召回率和 $F_{0.5}$ 分别超出 WEFR 对比方法53%、25%、43%。在未来的工作中将优化 LSTM 模型架构, 进一步改进所提方法, 提高检测性能。

参考文献:

- [1] 高聪明, 石亮, 刘凯, 等. 闪存固态硬盘系统结构与技术 [J]. 计算机研究与发展, 2021, 58(7): 1518-1532.
- [2] 王梓仲, 王海霞, 邵艾然, 等. 一种混合局部恢复码及 Hitchhiker 码的存储策略 [J]. 计算机学报, 2020, 43(4): 618-630.
- [3] NARAYANAN I, WANG Di, JEON M, et al. SSD failures in datacenters: what, when and why? [C]// Proceedings of the 2016 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Science. New York, NY, USA: ACM, 2016: 407-408.
- [4] XU Yong, SUI Kaixin, YAO R, et al. Improving service availability of cloud systems by predicting disk error [C]// Proceedings of the 2018 USENIX Conference on Usenix Annual Technical Conference. New York, NY, USA: ACM, 2018: 481-493.
- [5] GUNAWI H S, SUMINTO R O, SEARS R, et al. Fail-slow at scale: evidence of hardware performance faults in large production systems [J]. ACM Transactions on Storage, 2018, 14(3): 23.
- [6] 万洪浩, 邓明翥, 肖依, 等. 一种面向 RAID6 固态阵列的数据盘失效快速修复方法 [J]. 计算机研究与发展, 2015, 52(S2): 62-67.
- [7] WAN Honghao, DENG Mingzhu, XIAO Nong, et al. A fast recovery method for data disk failure in SSD-based RAID6 [J]. Journal of Computer Research and Development, 2015, 52(S2): 62-67.
- [8] HAN Shujie, LEE P P C, SHEN Zhirong, et al. Toward adaptive disk failure prediction via stream mining [C]// 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS). Piscataway, NJ, USA: IEEE, 2020: 628-638.
- [9] ZHANG Ji, WANG Yuanzhang, WANG Yangtao, et al. Tier-scrubbing: an adaptive and tiered disk scrubbing scheme with improved MTTD and reduced cost [C]// 2020 57th ACM/IEEE Design Automation Conference (DAC). Piscataway, NJ, USA: IEEE, 2020: 1-6.
- [10] XIAO Jiang, XIONG Zhuang, WU Song, et al. Disk failure prediction in data centers via online learning [C]// Proceedings of the 47th International Conference on Parallel Processing. New York, NY, USA: ACM, 2018: 35.
- [11] WANG Yufei, DONG Xiaoshe, WANG Longxiang, et al. Optimizing small-sample disk fault detection based on LSTM-GAN model [J]. ACM Transactions on Architecture and Code Optimization, 2022, 19(1): 13.
- [12] MAHDISOLTANI F, STEFANOVICI I, SCHROEDER B. Improving storage system reliability with proactive error prediction [C]// Proceedings of the 2017 USENIX Conference on Usenix Annual Technical Conference. New York, NY, USA: ACM, 2017: 391-402.
- [13] ALTER J, XUE Ji, DIMNAKU A, et al. SSD failures in the field: symptoms, causes, and prediction models [C]// Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. New York, NY, USA: ACM, 2016: 407-408.

- 2019: 75.
- [13] SARKAR J, PETERSON C, SANAYEI A. Machine-learned assessment and prediction of robust solid state storage system reliability physics [C]//2018 IEEE International Reliability Physics Symposium (IRPS). Piscataway, NJ, USA: IEEE, 2018: 3C. 6-1-3C. 6-8.
- [14] PULGAR F J, CHARTE F, RIVERA A J, et al. Choosing the proper autoencoder for feature fusion based on data complexity and classifiers; analysis, tips and guidelines [J]. *Information Fusion*, 2020, 54: 44-60.
- [15] 陈小红, 李霞, 王娜. 高维多目标优化中基于稀疏特征选择的目标降维方法 [J]. *电子学报*, 2015, 43(7): 1300-1307.
- CHEN Xiaohong, LI Xia, WANG Na. Objective reduction with sparse feature selection for many objective optimization problem [J]. *Acta Electronica Sinica*, 2015, 43(7): 1300-1307.
- [16] 宋昱, 孙文赞, 陈昌盛. 对数变换主成分分析的图像识别 [J]. *西安交通大学学报*, 2021, 55(1): 33-42.
- SONG Yu, SUN Wenyun, CHEN Changsheng. Log-arithmetic transformation based principal component analysis for image recognition [J]. *Journal of Xi'an Jiaotong University*, 2021, 55(1): 33-42.
- [17] DUAN Feng, REN Xina, YANG Yikang. A gesture recognition system based on time domain features and linear discriminant analysis [J]. *IEEE Transactions on Cognitive and Developmental Systems*, 2021, 13(1): 200-208.
- [18] POURRAMEZAN R, HASSANI R, KARIMI H, et al. A real-time synchrophasor data compression method using singular value decomposition [J]. *IEEE Transactions on Smart Grid*, 2022, 13(1): 564-575.
- [19] PAUL A, CHAKI N. Dimensionality reduction of hyperspectral image using signal entropy and spatial information in genetic algorithm with discrete wavelet transformation [J]. *Evolutionary Intelligence*, 2021, 14(4): 1793-1802.
- [20] CHEN Jiufang, YUAN Ye, RUAN Tao, et al. Hyper-parameter-evolutionary latent factor analysis for high-dimensional and sparse data from recommender systems [J]. *Neurocomputing*, 2021, 421: 316-328.
- [21] ZHAO Xiaoli, JIA Minping, LIU Zheng. Semisupervised deep sparse auto-encoder with local and nonlocal information for intelligent fault diagnosis of rotating machinery [J]. *IEEE Transactions on Instrumentation and Measurement*, 2021, 70: 1-13.
- [22] GUO Jiao, LI Henghui, NING Jifeng, et al. Feature dimension reduction using stacked sparse auto-encoders for crop classification with multi-temporal, quad-pol SAR data [J]. *Remote Sensing*, 2020, 12(2): 321.
- [23] PULGAR F J, CHARTE F, RIVERA A J, et al. ClEnDAE: a classifier based on ensembles with built-in dimensionality reduction through denoising autoencoders [J]. *Information Sciences*, 2021, 565: 146-176.
- [24] GAO Zhaoqi, LI Chuang, LIU Naihao, et al. Large-dimensional seismic inversion using global optimization with autoencoder-based model dimensionality reduction [J]. *IEEE Transactions on Geoscience and Remote Sensing*, 2021, 59(2): 1718-1732.
- [25] LIOU C Y, CHENG Weichen, LIOU J W, et al. Autoencoder for words [J]. *Neurocomputing*, 2014, 139: 84-96.
- [26] ANSARI M S, BARTOŠ V, LEE B. GRU-based deep learning approach for network intrusion alert prediction [J]. *Future Generation Computer Systems*, 2022, 128: 235-247.
- [27] XU Fan, HAN Shujie, LEE P P C, et al. General feature selection for failure prediction in large-scale SSD deployment [C]//2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). Piscataway, NJ, USA: IEEE, 2021: 263-270.

(编辑 武红江 刘杨)