Transfer Learning based Failure Prediction for Minority Disks in Large Data Centers of Heterogeneous Disk Systems

Ji Zhang[§], Ke Zhou[§],Ping Huang[§$],Xubin He[$],Zhili Xiao[ζ],Bin Cheng[ζ],Yongguang Ji[ζ],Yinhu Wang[ζ]

[§]Wuhan National Laboratory for Optoelectronics and School of Computer Science and Technology, HUST
[§]Key Laboratory of Information Storage System,Intelligent Cloud Storage Joint Research center of HUST and Tencent
[$]Temple University, [ζ]Tencent Inc.
{jizhang,k.zhou}@hust.edu.cn,{templestorager,xubin.he}@temple.edu
{tomxiao,bencheng,raidmanji,yhwang}@tencent.com

ABSTRACT

The storage system in large scale data centers is typically built upon thousands or even millions of disks, where disk failures constantly happen. A disk failure could lead to serious data loss and thus system unavailability or even catastrophic consequences if the lost data cannot be recovered. While replication and erasure coding techniques have been widely deployed to guarantee storage availability and reliability, disk failure prediction is gaining popularity as it has the potential to prevent disk failures from occurring in the first place. Recent trends have turned toward applying machine learning approaches based on disk SMART attributes for disk failure predictions. However, traditional machine learning (ML) approaches require a large set of training data in order to deliver good predictive performance. In large-scale storage systems, new disks enter gradually to augment the storage capacity or to replace failed disks, leading storage systems to consist of small amounts of new disks from different vendors and/or different models from the same vendor as time goes on. We refer to this relatively small amount of disks as minority disks. Due to the lack of sufficient training data, traditional ML approaches fail to deliver satisfactory predictive performance in evolving storage systems which consist of heterogeneous minority disks. To address this challenge and improve the predictive performance for minority disks in large data centers, we propose a minority disk failure prediction model named *TLDFP* based on a transfer learning approach. Our evaluation results on two realistic datasets have demonstrated that *TLDFP* can deliver much more precise results, compared to four popular prediction models based on traditional ML algorithms and two state-of-the-art transfer learning methods.

1 INTRODUCTION

Hard disks are widely used as the common and primary storage devices for large-scale storage systems in modern data centers. In such data centers, it has been an extremely challenging undertake to ensure high availability and reliability for IT management, as various disk failures constantly occur in the field, whether being hard disks [11, 21, 46] or flash-based SSDs [7, 19]. Disk failures can lead to temporary data loss and thus system unavailability or even permanent data loss if the lost data cannot be recovered by existing data protection schemes, e.g., replication and erasure codes [3, 9] due to disk failures exceeding the designed correction capability. A hard disk is a rather complex system consisting of a variety of magnetic, mechanical, and electronic components, each of which could fail. As a result, hard disk failures show different manifestations and extents of severeness [34] for numerous reasons, which has been observed in data centers from major IT companies [13, 29]. Compared with the traditional passive fault tolerance techniques like EC (Erasure Code) and RAID (Redundant Arrays of Independent Disks) [41], proactive disk failure prediction tends to ensure the reliability and availability of large-scale storage systems in advance. Therefore, successful disk failure prediction not only reduces the risk of losing data but also reduces the data recovery cost (i.e., network bandwidth) associated with recovering the data residing on failed disks.

Hard drive manufacturers implement the self-monitoring, analysis and reporting technology (SMART) technology [1] in the disk firmware. Most of the SMART attributes contain information about gradual degradations and possible defects of disks. Internally, a disk uses the so-called *“threshold method"* [39] based on SMART values to claim its failure status, which means the hard disk would raise an alarm if the value of an SMART attribute crosses the corresponding predefined threshold. However, this *“threshold method"* only achieves a failure detection rate (FDR) of 3%-10% with 0.1% false alarm rate (FAR) [39]. In other words, these numbers highlight the conservative nature of this method, i.e., it would rather miss chances to detect more disk failures than report false alarms at a higher rate.

To improve the predictive performance, several machine learning (ML) algorithms based disk failure prediction models [8, 26, 28] have been proposed, which leverage training SMART data to predict disk failures. Unfortunately, these works focused on a large number of homogeneous disks which have sufficient training data. In large-scale storage system scenarios, bunches of new disks enter gradually to replace failed disks, resulting in storage systems consisting of heterogeneous disks from different vendors and different models from the same vendor as time goes on. Heterogeneous disks

ICPP 2019, August 5–8, 2019, Kyoto, Japan

ACM ISBN 978-1-4503-6295-5/19/08…$15.00
https://doi.org/10.1145/3337821.3337881

with numerous disk models are common in data centers [22, 27]. Moreover, in evolving storage systems, some disk models are dramatically fewer than others and we call this relatively small amount of disks ***minority*** disks (conversely the large amount of disks as ***majority*** disks) in large data centers of heterogeneous disk systems. We found that about 25% of disks with numerous models (more than 50) are minority disks in two real-world data centers as detailed in Section 3.1. Due to the small sample and insufficient training data of minority disks, traditional ML algorithms using the training data of minority disks would dramatically increase the risk of over-fitting (Section 3.1) or poor generalization [48] which will weaken the performance of predictive models and seriously affect the reliability of the storage system. Therefore, we are poised to develop a disk failure prediction model *TLDFP* to predict failures for minority disks under the condition of having rich heterogeneous disk datasets. Our basic idea is to predict minority disk failures from the available majority disk datasets which is an application of transfer learning.

In this paper, we aim to seek answers to the following problems: **(1) What** is the definition of a minority disk dataset as far as failure predication is concerned? **(2) Why** should we use transfer learning for minority disks failure prediction? **(3) How** to use transfer learning methods to predict minority disks failure? **(4) When** to use transfer learning for minority disks failure prediction? Besides, when applied to two real-world datasets from the public Backblaze and Tencent which is one of the largest social network companies in the world, our method *TLDFP* achieves on average 96% failure detection rate with 0.5% false alarm rate when making cross-disk models failure prediction in addressing realistic system challenges.

2 BACKGROUND AND RELATED WORK

Almost all hard disk drives and flash-based SSDs come with built-in Self-Monitoring, Analysis and Reporting Technology (SMART), which are indicators of disk health status. The specification of SMART technology contains up to 30 attributes, reporting various disk operating conditions. SMART data directly or indirectly reflects the health condition of disks and even contains some statistical information. SMART data can be obtained through specified disk protocols upon which the disk manufacturer reached agreement. The hard disk would raise an alarm if the value of an SMART attribute crosses the corresponding predefined threshold. Each SMART attribute entry consists of five elements described as a tuple **(ID, Normalized, Raw, Threshold, Worst)**.

- **ID:** The designated sequence number of the SMART attribute.
- **Normalized:** Current or last normalized value (most are normalized to a value between the best value 253 and the worst value 1 calculated by manufacturer-specific algorithms using its raw value).
- **Raw:** The original value corresponding to counts or physical states provided by a sensor and vendor-specific.
- **Threshold:** The threshold value beyond which a disk alarms a failure.
- **Worst:** The lowest or worst value for a given attribute.

Not all five elements in a tuple are used. In our paper, we focus on the first three elements (ID, Normalized, and Raw) in our collected datasets. For convenience, we use **“smart_ID_Raw: V"** to denote the raw value of a SMART attribute whose ID is V. For example, **smart_1_Raw:10** means that the raw value of the read error rate attribute (ID:**1**) is 10 and **smart_5_Normalized:56** means that the normalized value of the reallocated sectors count attribute (ID:**5**) is 56. More specific information about the SMART attributes we use in our evaluation is given in Table 6.

There have proposed a host of ML algorithms for disk failure prediction models based on SMART data. Hamerly and Elkan [32] employ two Bayesian methods to model disk failure based on SMART data from Quantum Inc., which consists of 1,927 good drives and 9 failed drives. They categorize the problem as anomaly detection and establish a mixture model called NBEM, short for Naive Bayes clusters trained using expectation-maximization and another method called naive Bayes classifier. They achieve failure detection rates of 35-40% for NBEM and 55% for naive Bayes classifier at about 1% FAR. Hughes et al. [35] explore two statistical methods to improve predictive performance. They explore the capabilities of statistical tests like the rank sum test and OP-ed single variate test, and test both methods with 7,744 drives data (out of which 36 are failures) from two different disk models spanning across a period of 3 months. They achieved a failure detection rate (FDR) of 60% and 0.5% false alarm rate (FAR). Murray et al. [15] compare the predictive performance of SVM, rank-sum test, unsupervised clustering and reverse arrangements test.

Zhu et al. [8] explore the capability of a Backpropagation (BP) neural network and an improved SVM model to establish the prediction model based on SMART data. Many researchers use a Support Vector Machine (SVM [4]) because they claim SVM can efficiently perform a non-linear classification using the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces [44][31]. In order to improve the stability and interpretability of the disk failure prediction model, Li et al. [16] propose a new hard drive failure prediction model based on Classification And Regression Trees (CART). The Regression Tree can give the disk a heath assessment rather than a simple classification result. Gradient Boosted Regression Tree (GBRT [30]) has been proposed to model disk failure [17, 18], where GBRT is a gradient descent boosting technique based on tree averaging, and is an accurate and effective ML method that can used for both regression and classification problems. To avoid over-fitting, the GBRT algorithm trains many tree stumps as week learners, rather than full, high variance trees. Moreover, the Regularized Greedy Forests (RGF [36]) approach is a powerful, non-linear classification method. It is a variation of GBRT in which the structure search and optimization are decoupled and it utilizes the concept of structured sparsity to perform greedy search directly over the forest nodes based on the forest structure. Mirela Madalina Botezatu et al. employ this method to model disk failure and achieve good results [2]. Xu et al. [10] present a Recurrent Neural Networks (RNN [24, 25]) method to leverage sequential information for predicting hard disk failure. They use a dataset collected from a real-word data center containing 3 different disk models represented as W, S and M and establish the prediction model for those disk models, respectively. They model the long-term dependent sequential SMART data and demonstrate the capability of their predictive model. More recently, Mahdisoltani et al. [12] propose to use traditional ML algorithms to predict disk

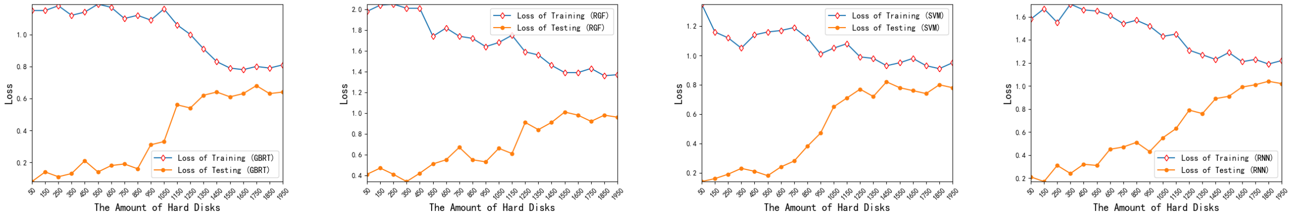


Figure 1: The training loss and testing loss of four popular traditional ML algorithms. Note that the y-axis means loss values as the dataset size increases.

sector errors using SMART datasets. Our goals in this paper are to make whole disk failure predictions which require much higher accuracy due to cost consideration.

As previously mentioned, the need for transfer learning occurs when there is a limited availability of training data from a new disk model, which regularly happens to evolving storage systems. With big data repositories becoming more prevalent, using existing datasets that are related to, but not exactly the same as, a target domain of focus point or interest makes transfer learning solutions an attractive approach. There are various applications in which transfer learning has been successfully applied to, including multi-language text classification [20, 43], image classification [6], human activity classification [33], text sentiment classification [14], Web document classification [45] and so on. Unsurprisingly, in recent years, researchers have started to use transfer learning method to solve minority disks failure prediction problems [2, 22]. Mirela Madalina Botezatu et al. proposed the sample selection de-biasing method [2], which we denoted as *SSDB* in our paper. Its main idea is to train a classifier that can rank the observations linked to a specific disk model based on their similarity to samples pertaining to the target disk model. This method is also a single-source domain transfer learning method in spirit similar to *TLDFP* algorithm. FLF Pereira et al. proposed the multi-source domain transfer learning for Bayesian network [22], which we denoted as *TLBN* in our paper. It proposes a new source building method called clustering-based information source and groups several HDDs according to their similarity to build a novel information source for transfer learning. Although these methods also provide a solution to minority disks failure prediction, we have compared *TLDFP* with them and show our approach delivers better predictive performance. Besides, note that our work is the first to systematically (**What, Why, How and When**) propose using the transfer learning method to solve minority disks failure prediction based on SMART attributes for large-scale, active, evolving storage systems.

3 PRELIMINARY STUDY AND MOTIVATION

In this section, we define minority disk datasets via experimental examination, investigate the distributions of SMART data, and justify **why** we use transfer learning for minority disk failure prediction.

3.1 Minority Disk Datasets

As mentioned previously, we aim to improve disk failure predictive performance for a disk dataset which has insufficient training data and where traditional ML algorithms deliver suboptimal performance. In this section, we give the definition of a *Minority Disk Dataset* and quantitatively evaluate them via experimentally showing the *training loss* and *testing loss* [23] of four popular ML algorithms (GBRT, RGF, SVM, and RNN) in disk failure prediction [2, 10, 17, 31]. A loss is a number indicating how bad a model's prediction. Note that we have added a regularization term to construct the loss function in all four methods which can effectively prevent over-fitting caused by the model having a very large number of parameters. Figure 1 illustrates the results. As can be seen from the figure, with the dataset increasing, the loss of training set increases to a certain extent while the loss of testing set decreases because the increase in data set leads to more complex situations where the training model needs to be fitted. More specifically, when the amount of disks is less than 1500, the gap between the loss of training and testing decreases as the data set enlarges, which is called *over-fitting* caused by minority disks. When the amount of disks goes over 1500, the gap becomes smaller and stabilizes. Therefore, we can draw the following conclusions: **(1)** A disk dataset containing fewer than 1500 disks could lead to over-fitting which we name it as *Minority Disk Datasets* ; **(2)** The four popular traditional ML algorithms cannot deliver satisfactory performance when the dataset contains fewer than 1500 disks. As far as we know, we are the first to define minority disk datasets and quantitatively evaluate them through extensive data analysis and experiments. We studied two real data centers and categorized the disk quantities by the threshold of 1500. As shown in Table 1, in data center BackBlaze, **91** different disk models only account for less than **24%** of all disks while 12 models account for more than 76%. We call these 91 models minority disks. A similar observation has been found in data center of Tencent.

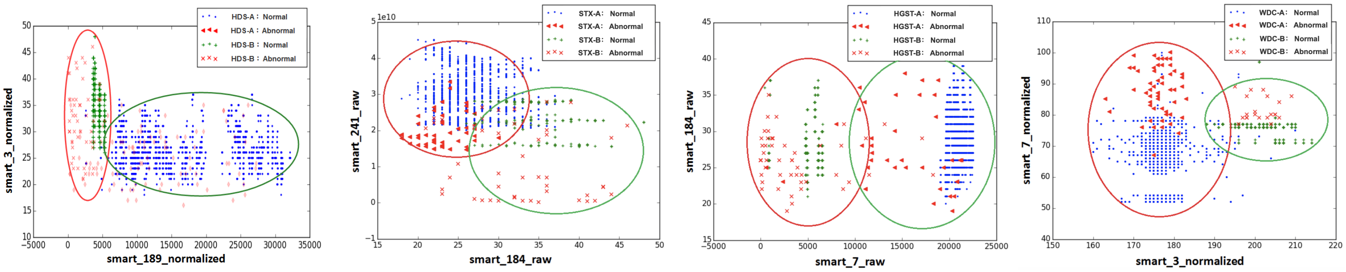
The above description and analysis implies that making disk failure prediction for minority disks is a realistic problem that needs to be resolved.

Table 1: Characteristics of disk population

Data Center	Disk Number	Disk Models	Total Number	Percentage
Backblaze	≥ 1500	12	114,570	76.61%
	< 1500	**91**	34,978	**23.39%**
Tencent	≥ 1500	8	18,996	73.32%
	< 1500	**52**	52,235	**26.67%**

3.2 The Baseline Results of using Traditional ML only Trained on Minority Disk Datasets

In order to investigate the predictive results of using traditional ML methods **only** trained on minority disk datasets, we use four minority disk models from four disk manufacturers to conduct this experiment based on four popular traditional ML methods: GBRT, RGF, SVM and RNN which include the popular tree structure algorithms and deep learning algorithms and have been commonly used in disk failure prediction. Note that we only use 70% of the minority disk datasets as training sets and the remaining 30% as testing sets without any other datasets. Besides, we consistently use the following acronyms for these four vendors throughout the



(a) HDS (b) STX (c) HGST (d) WDC

Figure 2: The distributions of two SMART attributes of two disk models from four manufacturers, i.e., Hitachi, Seagate, HGST, WDC. Each subfigure shows the *Abnormal* and *Normal* states indicated by a randomly chosen pair of SMART attributes of two disk models. These four subfigures indicate that two disk models of each manufacturer exhibit similar failure patterns represented by the two SMART attributes distributions and the SMART data are distributed in different value ranges, which motivates us to apply transfer learning to make cross-model disk failure predictions.

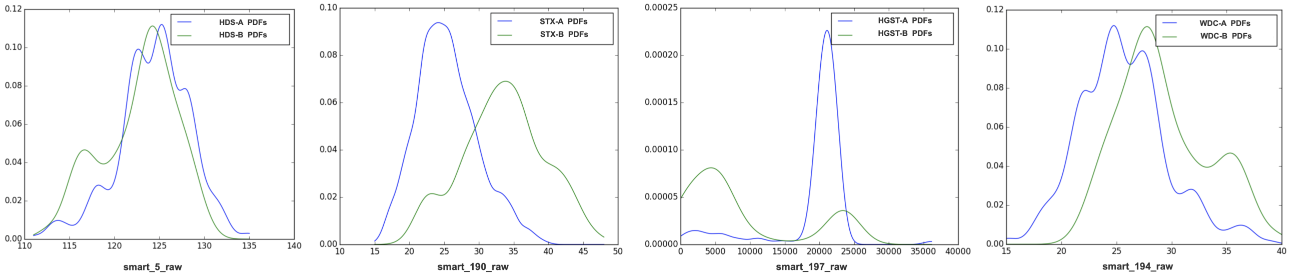


Figure 3: PDFs of a SMART attribute value of two different disk models from four manufacturers.

paper which are also used in their disk models: Hitachi(HDS) and Seagate(ST) come from Backblaze, Hitachi Global Storage Technologies(HGST) and Western Digital(WDC) come from Tencent. As can be seen from the Table 2, none of the four traditional ML methods can deliver a high FDR and low FAR (FDR and FAR see Section 5.1.2 in detail). We know the poor predictive performance due to over-fitting caused by using small homogeneous datasets based on traditional ML.

Table 2: The predictive results of minority disk failure prediction via traditional ML

Methodology	Manufacturer	FDR	FAR
GBRT	HDS/STX/HGST/WDC	27.3%/37.5%/31.6%/38.5%	29.0%/19.4%/17.6%/21.8%
RGF	HDS/STX/HGST/WDC	36.4%/50.0%/47.4%/53.8%	44.3%/22.4%/ 53.4%/36.6%
SVM	HDS/STX/HGST/WDC	50.0%/41.7%/57.9%/30.8%	20.7%/47.8%/ 24.0%/43.7%
RNN	HDS/STX/HGST/WDC	40.9%/33.3%/36.8%/30.8%	28.3%/31.3%/ 39.7%/38.7%

We have also conducted experiments to directly use large datasets of the available majority disks to predict minority disk failures based on the four popular traditional ML techniques, the performance is not satisfactory either (details are shown in Figure 5 in Section 5). To understand the reason, we analyze the SMART data distributions as below.

3.3 SMART Data Distributions

It is interesting to observe that the values of SMART attributes indicating disk health conditions of different disk models from the same manufacturer exhibit similar distribution patterns. We have analyzed both the publicly available SMART dataset from Backblaze [1] and a dataset from the datacenter of Tencent. Figure 2 shows the revealed SMART data distribution patterns. Each subfigure shows a pair of SMART attribute values' distribution pattern of two different disk models from the same manufacturer and each circle is used to highlight different disk models. As it is evidently shown, the relationship between *Abnormal* and *Normal* states indicated by the two SMART attributes of two disk models shows a similar pattern, with only the difference of SMART values being in different ranges. Figures 2(a), 2(b) , 2(c) and 2(d) respectively show that the *Abnormal* state is above, below, and to the left of the *Normal* state for the two disk models from the same manufacturer. Furthermore, the SMART values are distributed in different spectrums. Take Figure 2(b) Seagate as an example, the distribution region of model STX-B is right-lower than that of model STX-A. Traditional ML algorithms deliver good predictive performance only when both training and testing data are drawn from the same distribution [40]. Therefore, they fail to perform satisfactorily when it comes to cross-disk models failure prediction due to different distribution spectrums as revealed in Figure 2.

In order to take an in-depth look at the distributions of SMART values of different disk models from the same manufacturer and further motivate the transfer learning from one disk model to a different model and explain why we use transfer learning for minority disks failure prediction, we investigate the differences in the distributions of SMART data and present a intuitive analysis comparing different disk models from the same manufacturer. Probability Density Function statistic (PDFs) is frequently used to describe the intensity of continuous random variable. For easy observations, we use the Gaussian Kernel Density Estimation (GKDE) as the kernel function, which results in smooth curves.

Figure 3 shows the PDFs of the value of a SMART attribute of two models from four manufactures. The distributions of the SMART data of two disk models are different but similar in that they show similar spikes though at different points and magnitudes. We refer to this phenomenon as *covariate shift* [47] among relevant predictors

[1] https://www.backblaze.com/b2/hard-drive-test-data.html

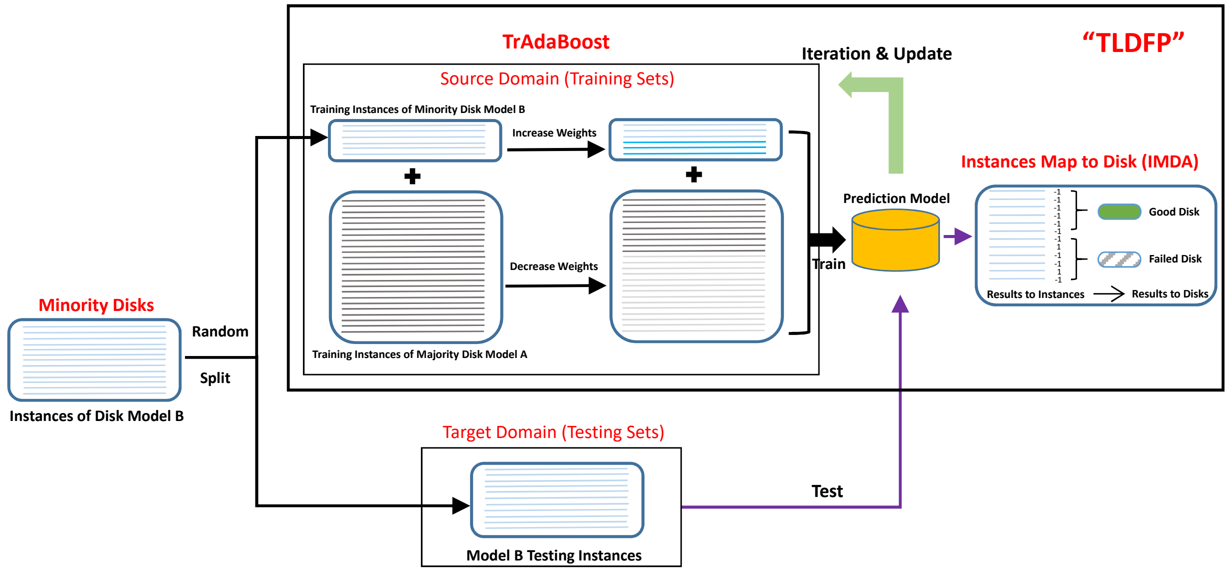

TrAdaBoost
Iteration & Update
"TLDFP"
Source Domain (Training Sets)
Training Instances of Minority Disk Model B
Increase Weights
Decrease Weights
Training Instances of Majority Disk Model A
Prediction Model
Train
Instances Map to Disk (IMDA)
Good Disk
Failed Disk
Results to Instances
Results to Disks
Minority Disks
Random
Split
Instances of Disk Model B
Target Domain (Testing Sets)
Model B Testing Instances
Test

Figure 4: The overall structure of *TLDFP*, which contains the transfer learning algorithm *TrAdaBoost* and the *Instances Map to Disk Algorithm (IMDA)*. Source domain contains a fully labeled dataset of majority disk model A and a small portion of labeled dataset of minority disk model B. The testing data in the target domain is the remaining unlabeled dataset of minority disk model B.

between different models from the same manufacturer. Therefore, we conclude that different disk models from the same manufacturer exhibit varying SMART value distributions. For the problem of minority disks failure prediction, the implication is that a prediction model built upon traditional ML algorithms using training data from one disk model is not applicable to other different models even from the same manufacturer. Therefore, to leverage a prediction model for a disk model built on adequately sufficient SMART training data to build a predictive model for a different model for which there is only limited training data, we could employ transfer learning algorithm which is inherently suitable for transferring health state information from one disk model to another disk model from the same manufacturer.

Considering the above regularity of relationship between *Abnormal* and *Normal* disk states and varying SMART data distribution spectrums across different disk models, we are motivated to apply transfer learning to predict disk failures for minority disks using the knowledge from the majority disks, which we name as *TLDFP*.

4 MINORITY DISK FAILURE PREDICTION

In this section, we answer the questions of **how and when** to use transfer learning for minority disk failure prediction. Specifically we detail our transfer learning method for minority disks failure prediction *TLDFP*, followed by our method of selecting source domain based on KLD.

4.1 *TLDFP*: Transfer Learning for Minority Disk Failure Prediction

We elaborate **how** to use transfer learning to predict minority disk failure in this section. Figure 4 illustrates the overall structure of our proposed predictive method *TLDFP*. It consists of two major components: a transfer learning algorithm *TrAdaBoost* [5] and an instance map to disk algorithm *IMDA*. Note that we randomly divide the SMART data of minority disk model B into two parts. The first part includes a small portion (e.g., 10%) of the labeled target domain data, which is then put together with the data of majority disk model A as a combined source domain to establish the relationship between the two different disk models so as to reduce variation between their distributions. The other part contains the remaining unlabeled data as testing data. Then we use our *TLDFP* method to establish a predictive model and make failure prediction for the minority disk model B training data. With the above description, the problem we aim to solve in this paper can then be formally defined as: given enough labeled training data S_a, a small amount of labeled training S_b and unlabeled testing data T_b, the main objective is to leverage the useful portions of S_a and S_b and train a classifier C which achieves a good performance of classifying the unlabeled training set T_b. The *TrAdaBoost* is an extension of the traditional ML method *AdaBoost*. *AdaBoost* is an iterative algorithm and its key procedure includes training several different weak classifiers with different weights and then consolidating those weak classifiers to a strong classifier to boost predictive performance. According to the *AdaBoost* algorithm, it first gives an initial weight to all training instances. When an instance in the source domain is found to be misclassified, we consider this instance as difficult to classify and thus increase its weight. In this way, the significance of this instance will become greater in the next iteration. However, *AdaBoost* is a traditional ML method that can only build effective predictive model for testing data which has the same distribution as the training data. In the transfer learning algorithm *TrAdaBoost*, when an instance of disk model B from the combined source domain is misclassified, we increase the weight of this instance in the next iteration, which is similar to *AdaBoost*. However, when an instance of disk model A is mispredicted, the instance is assumed to be different from disk model B. Therefore, unlike *Adaboost*, it decreases the weight of that instance in the next iteration to reduce its influences on the target domain.

The input of the *TrAdaBoost* algorithm includes two disk models' training and testing data, and the maximum number of iterations T.

It initializes the weights of training data and performs the iteration process. In each cycle, we use the basic learner, such as GBRT, RGF, SVM, RNN, and the weight distribution I^t to build a classifier h_b on testing data and calculate the error rate on the labeled source domain disk model B data set S_b. Lastly, we set the new weights based on the previous iteration results and the error rate. Note that if majority disk model A instances of the source domain are misclassified, they are considered to be different from the minority disk model B. As a result, we reduce the weights of these instances in order to reduce their influences on the predictive model in the next iteration. Specifically, we multiply the instances by $w_j^t \varphi^{|c(x_j)-h_t(x_j)|}$, where φ ranges from 0 to 1 and $c(\cdot)$ is the true label of a SMART attribute. On the other hand, if the disk model B instances in the combined source domain are misclassified, we increase the weights of these instances to gain more attention in the next iteration through multiplying these instances by $w_j^t \varphi_t^{|c(x_j)-h_t(x_j)|}$, where φ_t is greater than 1. After several iterations (we set the maximum number of iterations to 22, though we will investigate how this value affects the predictive performance in Section 6.4), the instances of majority disk model A in the source domain that fit minority disk model B will gain greater weights and those different from disk B will have smaller weights.

Since the inputs of the failure prediction model include many SMART instances from a lot of disks at different moments, each output result indicates the prediction result for a particular instance rather than the disk health state. Therefore, we need to map the results of multiple SMART instances to the final disk state. To achieve that, we propose an *Instances Map to Disk Algorithm* (*IMDA*). *IMDA* determines the final health state of a disk by considering all of its SMART instances. Specifically, if any instance is classified as failure, the corresponding disk is considered as failure and other alternative options are discussed in Section 6.5.

4.2 Source Domain Selection based on KLD

When can we use *TLDFP* for minority disks failure prediction? To answer this question, we use Kullback Leibler Divergence (KLD), which is a metric measuring the divergence degree of one probability distribution from another expected probability distribution [38]. KLD values indicate the disparities between two random variable distributions. A zero KLD value means that the two random distributions are the same, while the KLD value increases as the differences between two random distributions widen. In general, the bigger a KLD value is, the greater differences between two distributions will be and the more difficult the knowledge transfer between two distributions will be. Table 3 gives KLD values corresponding the PDFs showed in figure 3. Table 3 shows that all KLD values are not equal to zeros, confirming that the respective SMART data distributions are indeed not the same. Note that as indicated in Table 3, the KLD value trend, which is consistent with the PDF differences increase from *HDS*, to *STX*, to *HGST*, to *WDC* shown in Figure 3. Considering that *TLDFP* is a method to decrease the data distribution differences between source domain and target domain, so we infer that the bigger KLD value between one disk model and another is, the harder *TLDFP* can transfer experience. The predictive results in Section 5.2 proves our conjecture and we also conduct detailed experiments and discussions in Section 6.1. Therefore, the value of KLD can guide us to select appropriate majority disk dataset (source domain) for training the minority disk failure predictive model. As far as we know, we are the first attempt to present a novel method based on KLD values as an effective indicator to select proper majority disk models and improve disk failure prediction. Our evaluation results in Section 6.1 demonstrate that our approach of using KLD is very effective and practical.

Table 3: The KLD values of the PDFs in Figure 3

Source Domain	Target Domain	SMART Attribute	KLD
HDS-A	HDS-B	5_RAW	0.61
STX-A	STX-B	190_RAW	0.89
HGST-A	HGST-B	197_RAW	1.35
WDC-A	WDC-B	194_RAW	0.56

5 EXPERIMENTAL EVALUATION

In this section, we evaluate the predictive performance of *TLDFP*. We first describe the methodology, followed by the experimental results of comparing *TLDFP* against four ML algorithms and two state-of-the-art transfer learning methods according to the evaluation metrics.

5.1 Methodology

We describe the characteristics of two real-world SMART datasets in our experiments and SMART attributes selection. Then we introduce four evaluation metrics commonly used in ML and some testing methods we use to conduct all our experiments.

5.1.1 *Datasets and Attribute Selections.*

Datasets: We use two SMART datasets from real-world data centers for evaluations. Table 4 gives the overall characteristics of the two datasets. Every disk is classified either as "Good" or "Failed". "Sample" indicates SMART records. Each good disk or failed disk has many SMART records. As the original data set has more samples of good disks than failure disks, we use majority class under-sampling to improve training in the case of imbalanced classes to create training datasets. We have chosen a 1:3 ratio of failure disk to good disk [12]. When performing the training in traditional ML method, we divide the data into **70% training and 30% testing data** for our experiments which is in line with existing work [22]. Note that all the results of our experiments are obtained by **cross-validation** [37] in order to avoid fortuitous accident which is common used in ML. Table 5 lists our chosen disks for evaluations.

Table 4: SMART Datasets

Data center	Duration	Good	Good Sample	Failed	Failure Sample
BackBlaz	50 months	141,891	106,867,099	7,657	7,689
Tencent	26 months	68,436	774,994,430	2795	31,574,341

Table 5: Selected disk models used in evaluations

Data center	Manufacturer	Disk model	Good	Bad
Backblaze	Hitachi	HDS722020ALA330	4774	225
		HDS723030ALA640	1048	72
	STX	ST4000DM000	37006	3157
		ST4000DX000	222	81
Tencent	HGST	HGST-A	13367	451
		HGST-B	679	63
	WDC	WDC-A	6847	259
		WDC-B	472	42

SMART Attribute Selections: Each SMART observation can contain up to 30 meaningful SMART attributes. However, some attributes are irrelevant to our disk failure predictive model because they are immutable or have not experienced noticeable abnormal changes. Therefore, we selectively keep those attributes that are relevant to the disk health state according to a feature selection process based on Principle Component Analysis (PCA) while ignoring other irrelevant attributes. The selected SMART attributes are listed in Table 6. For each SMART sample, we use the *Normalized Value* and *Raw Value*. The normalized value typically represents the current value of an attribute. However, certain normalized values lose accuracy when transformed from the raw value and some raw values are more sensitive to the predictive model. We also use other methods of feature selection like [2, 12, 22]. However, the impact on the experimental results is not significant, so we do not discuss further due to limited space.

In addition, different SMART attributes have different output ranges, which will lead to different impacts on the predictive model. In order to make a fair comparison among different SMART attributes in our disk failure predictive model, we normalize the range of all selected SMART attributes using the min-max scaling which is in line with existing work [12]:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

where x is the original value of a SMART attribute, x_{max} and x_{min} are the maximum and minimum value of the attribute in the training set, respectively.

Table 6: SMART attributes selected for our evaluations

#ID	SMART Attribute Name	Attribute type
001	Raw Read Error Rate	Normalized&Raw
003	Spin-Up Time	Normalized
005	Reallocated Sectors Count	Normalized&Raw
007	Seek Error Rate	Normalized&Raw
009	Power-On Hours	Normalized&Raw
184	I/O Error Detection and Correction	Normalized&Raw
187	Reported Uncorrectable Errors	Normalized&Raw
188	Command Timeout	Raw
189	High Fly Writes	Normalized&Raw
190	Airflow Temperature	Normalized&Raw
193	Load/Unload Cycle Count	Normalized&Raw
194	Temperature	Normalized&Raw
197	Current Pending Sector Count	Normalized&Raw
240	Head Flying Hours	Raw
198	Offline Uncorrectable Sector Count	Normalized&Raw
241	Total LBAs Written	Raw
242	Total LBAs Read	Raw

5.1.2 ***Evaluation Metrics.*** We use the following four metrics to report the results in our experiments which are commonly used for evaluating the capability of a classification model in ML [42].

FDR: *Failure Detection Rate*(FDR) also called *recall rate*. It captures the proportion of true failed disks that are correctly predicted as failed. The higher the FDR is, the better the model is.

FAR: *False Alarm Rate*(FAR), the proportion of good disks that are falsely predicted as failed. The lower the FAR is, the better the model is.

F-Score: F-Score is a balance between the two metrics FDR and *Prediction Precision* (PP). PP is the proportion of predictive failed disks that are correctly predicted as failed. The higher the F-Score is, the better the model is.

AUC-ROC Curve: The *Area Under the Curve-Receiver Operating Characteristic*(AUC-ROC) curve is a performance measurement for classification problem at various threshold settings. ROC is a probability curve and AUC represents degree or measure of separability. It is plotted with FDR against the FAR where FDR is on y-axis and FAR is on the x-axis. In disk failure prediction, a higher the AUC means the model is better at distinguishing failed and good disks.

5.1.3 ***Testing Methods and Configurations.*** To verify the effectiveness of our proposed *TLDFP*, we conduct experiments in three scenarios: 1) to use traditional ML methods only trained on the minority disk datasets, 2) to compare *TLDFP* with traditional ML techniques (**baseline**), and 3) to compare *TLDFP* with other transfer learning approaches. The settings are described below.

1) Traditional ML methods only trained on minority disk: The detailed description see Section 3.2

2) ***TLDFP*** **with traditional ML techniques:**

In this scenario, we conduct experiments to investigate the performance of minority disks failure prediction based on four traditional ML algorithms using large heterogeneous datasets for training from both different disk models and the same disk manufacturer. Table 7 shows the training and testing datasets. Note that we randomly choose 10% testing datasets for training with all training datasets.

3)***TLDFP*** **with other transfer learning approaches:**

In addition to traditional ML techniques, we also compare our *TLDFP* with two state-of-the-art transfer learning methods (*SSDB* and *TLBN*) of predicting minority disks failure. Note that we use the same datasets for the two methods in [2] and [22] for fair comparison in all our experiments.

5.2 Experimental Results

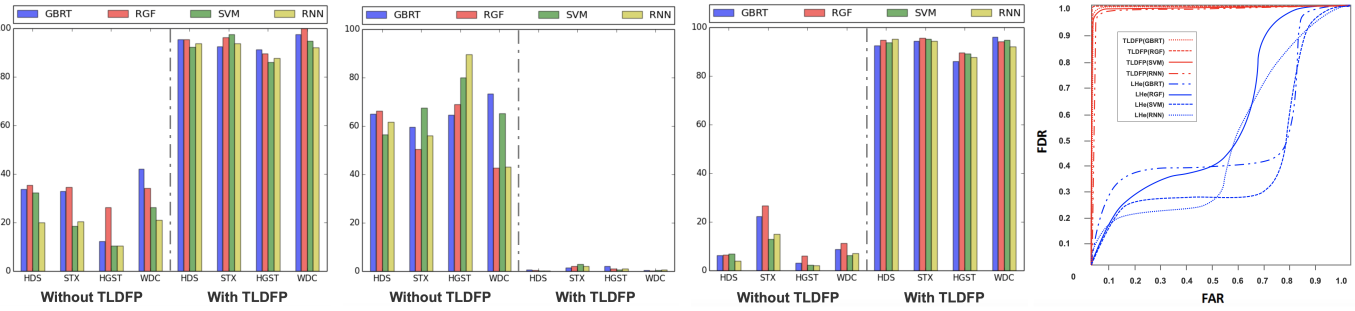
In this section, we show the results of the *TLDFP* compared to traditional ML methods and other transfer learning methods with four evaluation metrics mentioned in Section 5.1.2 respectively. **Note that we had showed the poor baseline results of using traditional ML methods only trained on the minority disk datasets in Section 3.2**.

5.2.1 ***Evaluations Compared to Traditional ML Approaches.***

- **FDR/Recall Rate:** We conduct experiments to investigate the FDR of *TLDFP* and four popular traditional ML methods using four disk models from two real data centers. As can be seen from the figure 5(a), none of the four traditional ML methods can deliver a high FDR using large heterogeneous datasets. However, the *TLDFP* use the above GBRT, RGF, SVM, RNN algorithms respectively as the basic learners all achieved higher FDR.
- **FAR:** Note that the goal of our *TLDFP* is not only to achieve high FDR but also low FAR for minority disk failure prediction. The results of FAR are showed in Figure 5(b). All other methodologies show higher FAR which is unacceptable in realistic data centers. Further, none of the four traditional ML methods can deliver both a high FDR and a low FAR on minority disks except for *TLDFP*. Based on the analysis in Section 3.3, we know the poor predictive performance caused by the traditional ML methods do not have the ability to reduce the distribution difference between the minority

Table 7: Datasets of minority disk failure prediction using large heterogeneous dataset based on traditional ML

Data Center	Manufacturer	Training Disk Model	Testing Disk Model	Training set	Testing set
Backblaze	Hitachi	HDS-A	HDS-B	4774 good HDS-A and 225 failed HDS-A 105 good HDS-B and 7 failed HDS-B	943 good HDS-B and 65 failed HDS-B
	STX	STX-A	STX-B	37006 good STX-A and 3157 failed STX-A 22 good STX-B and 8 failed STX-B	200 good STX-B and 73 failed STX-B
Tencent	HGST	HGST-A	HGST-B	13367 good HGST-A and 451 failed HGST-A 68 good HGST-B and 6 failed HGST-B	611 good HGST-B and 57 failed HGST-B
	WDC	WDC-A	WDC-B	6847 good WDC-A and 259 failed WDC-A 47 good WDC-B and 4 failed WDC-B	425 good WDC-B and 38 failed WDC-B



(a) FDR(The higher, the better)

(b) FAR(The lower, the better)

(c) F-Score(The higher, the better)

(d) AUC-ROC

Figure 5: The results of FDR, FAR, F-Score and AUC-ROC using four disk models based on *TLDFP* compared to four traditional ML methods.

disk datasets in target domain and majority disk datasets in source domain.

- **F-Score:** Figure 5(c) compares the F-Score of the different prediction models on the two datasets using four disk models. As can be seen, *TLDFP* has much higher F-Score than other different traditional ML methods. As an example, the F-Score of *TLDFP* with RBF as its basic learner *TLDFP(RBF)* is almost 9 times as high as the algorithm RBF for WDC disks from data center of Tencent. As we recall in Section 4.2, the bigger KLD value of HGST dataset will lead to more difficult knowledge transfer. This conclusion is also confirmed by the F-Score observations given that **the F-Score of *TLDFP* for HGST are generally lower that other cases**. We will further discuss this issue in detail in Section 6.1.
- **AUC-ROC Curve:** We plot the AUC-ROC curve in Figure 5(d) using WDC disk model in Tencent. As it is shown, the AUC-ROC curve of *TLDFP* are all close to the top left corner and *TLDFP(RGF)* achieving the higher AUC value compared to other two transfer learning methods. The four traditional ML methods achieved lower AUC values, reflecting their poor classification ability in performing cross-disk model failure predictions.

5.2.2 *Evaluations Compared to Other Transfer Learning Approaches*.

As it is shown in Table 8, *TLDFP* shows higher FDR/F-Score/ and lower FAR than *SSDB* and *TLBN*. The reason is that although *SSDB* matches the distribution of the source domain with the target domain, it only ranks the observations in source domain, while *TLDFP* makes more effective weight adjustments to every observation. Considering the *TLBN* is a multi-source domain transfer learning method and *TLDFP* is a single-source domain transfer learning method, we analyze the KLD values between each disk model in the source domain and the minority disk model in the target domain. The results are showed in Table 9. We find the data in the disk model with a large KLD value in the source domain (such as ST320005XXXX and ST1500DL003). However, *TLDFP* only uses the disk model which has the smallest KLD value as source domain. A large KLD value leads to difficulties for *TLBN* in mitigating the distribution differences between the source and target domains. This result also shows that single-source domain transfer learning performs better than multi-source domain transfer learning and there is a good metric(e.g, KLD) for evaluating differences between different domains. **Note that we don't include the results of all methods and disk models due to space limit. From all our tests, *TLDFP* demonstrates the best performance.**.

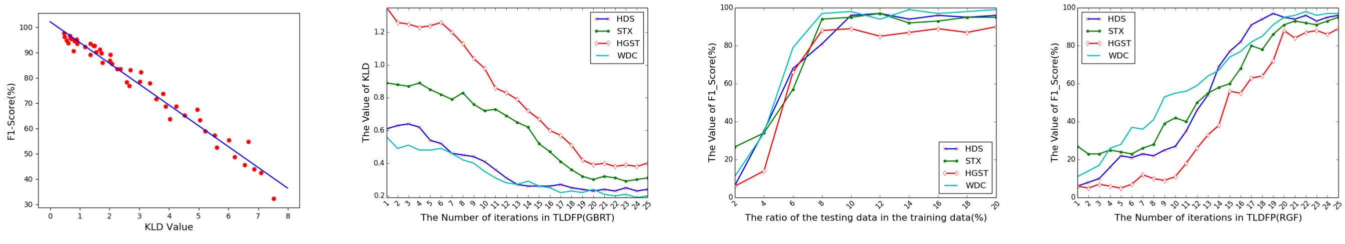
Table 8: The FDR, FAR, F-Score and AUC of the comparisons between *TLDFP* and *SSDB*, *TLBN*

Methods	Manufacturer	FDR	FAR	F-Score	AUC
TLDFP(RGF) VS SSDB	STX	94.9%/85.8%	1.6%/3.0%	92.6%/83.7%	0.93/0.86
	Hitachi	97.1%/70.8%	0.9%/5.4%	95.7%/69.0%	0.96/0.81
TLDFP(RGF) VS TLBN	STX	91.3%/73.1%	0.6%/2.6%	91.3%/70.4%	0.91/0.83

Table 9: KLD values between each disk model in the trainingsets and the minority disk model in the testingset using method *TLBN*

Training Disk Model	Testing Disk Model	SMART Attribute	KLD
ST320005XXXX	ST33000651AS	5_RAW	5.6
ST32000542AS		190_RAW	2.7
ST1500DL003		188_RAW	7.1
ST31500341AS		190_RAW	1.5
ST31500541AS		197_RAW	0.83

In summary, the results have demonstrated that *TLDFP* can effectively solve the problem of minority disk failure prediction with much better predictive performance than traditional ML methods and two other transfer learning methods for the same datasets. More specifically, *TLDFP* not only delivers high FDR, F-Score, AUC, but also shows a rather low FAR at the same time. The main reason for the comparison results is that our *TLDFP* algorithm is able to utilize the small number of labeled target disk data to establish the relationship between source and target disk models, which helps the large heterogeneous disk model to be well trained toward the characteristics of the minority target disk model. In other words, *TLDFP* reduces the difference in data distribution between the source and target domain, which we will further discuss in Section 6.2. Note that we had already verified the *TLDFP* performance on SSD and NVMe and collected some promising results, which are not included due to the space limit.



(a) (b) (c) (d)

Figure 6: (a) The fitting curves of KLD and F-Score. (b) KLD value decrease in training. (c) The relationship between FDR and percentage of target domain data drawn into source domain. (d) The relationship between FDR and the number of iterations in the *TLDFP*.

Table 10: The F1-Score varies with the value of KLD

Data Center	Method	Training Model	Testing Model	KLD	F1-Score
BackBlaze	*TLDFP(GBRT)*	HGST-K	HGST-L	2.67	76.8%
			HGST-M	1.76↓	85.9%↑
			HGST-N	0.91↓	93.5%↑
			HGST-O	0.69↓	95.7%↑
			HGST-P	0.47↓	97.6%↑
Tencent	*TLDFP(SVM)*	STX-K	STX-L	3.57	71.6%
			STX-M	2.58↓	78.2%↑
			STX-N	2.26↓	83.4%↑
			STX-O	1.35↓	93.1%↑
			STX-P	1.17↓	92.2%↓
			STX-Q	0.71↓	95.3%↑
			STX-R	0.66↓	96.6%↑

6 OBSERVATIONS AND SENSITIVITY STUDY

In this section, we provide several additional sensitivity studies from five aspects.

6.1 The Impact of KLD in Source Domain Selection

In order to verify our conjecture in Section 4.2 and further explain the results in Section 5.2.1 and Section 5.2.2, we analyze the relationship between KLD and F-Score in *TLDFP* using several minority disk models as testing disk model and large heterogeneous datasets of one disk model for training from the same manufacturer in two real data centers. The results are shown in Figure 6(a) and Table 10. We observed that the value of F-Score keeps rising as the value of KLD decreases. In other words, it shows that the smaller the difference in SMART data distribution between the source and target domain, the easier knowledge transfer in *TLDFP* can be. Therefore, we use KLD as an effective indicator for source domain selection of large heterogeneous dataset and usually select the one which has the smallest KLD value.

6.2 The Change of KLD Value in *TLDFP*

In order to more intuitively observe how the *TLDFP* method reduces the difference in data distribution between the source domain and the target domain, we record the value of KLD between the training set and the dataset after each iteration of the model during the training process of *TLDFP(GBRT)*, as illustrated in Figure 6(b). We can see that as the number of iterations of the model increases, the KLD value between the source domain and the target domain decreases continuously, and stabilizes at a smaller value when the number of iterations is about 22. This shows that our *TLDFP* model continuously reduces the difference of the SMART data distribution between the two domains in the training process, enabling us to use the large heterogeneous disk data to predict the minority disk data and realize knowledge transfer.

6.3 Varying Samples from Target Domain

As discussed previously our prediction model *TLDFP* uses a portion of labeled dataset from target domain as part of its source domain dataset. In this section, we investigate how the percentage number affects predictive performance of *TLDFP*. We report the results of *TLDFP* with RGF as its basic learner and using the disk data of WDC model from Tencent data center, as the other three basic learners show similar results. Figure 6(c) shows the relationship between the FDR and the percentage of target domain data put in the source domain. As it clearly shows, the FDR increases dramatically when the percentage increases from 2% to 10%. When the percentage goes beyond 10%, the FDR does not continue to increase but remains a relatively high level, meaning putting more target domain data to the source domain does not help further improving predictive performance. Consequently, we randomly choose 10% target domain data in our previous experiments.

6.4 The Impact of Iterations

The *TrAdaBoost* algorithm takes an input parameter to cap the iterations performed by the algorithm. In this section, we study how the iteration parameter affects predictive performance in terms of F-Score. We report the results of *TLDFP* with RGF as its basic learner. Figure 6(d) shows how F-Score changes as the number of iterations varies for different datasets. From this figure, we can make similar observations as in the preceding subsection. As the number of iterations increases, the F-Score increases quickly and reaches a stable level at 22 iterations. It also shows that as the algorithm adjusts instance weights in each iteration, *TLDFP* gradually adjusts to converge toward the testing data. Since the number of 22 represents a reflection point, we adopt this iteration number in our previous experiments.

6.5 The Sensitivity Study of IMDA

In Section 4.1, we introduced the algorithm IMDA. Specially, if any instance is classified as failure, the corresponding disk is considered as failure. Here we conduct experiments to evaluate the impact of different instances, specifically, 1 instance, 1/3 of all instances, 1/2 of all instances, 2/3 of all instances, and all instances being classified as failure.

The result of this experiment using *TLDFP(RNN)* based on the WDC disk model data in Tencent is showed in Table 11. It is clear that the option we use (one instance failure indicates the corresponding disk failure) achieves the best performance.

Table 11: The sensitivity study results of IMDA in *TLDFP*

Methods	Metrics	1	1/3	1/2	2/3	All
TLDFP	FDR	95%	32%	24%	8%	3%
	FAR	0.7%	0.7%	0.5%	0.2%	0.2%

7 CONCLUSION

In this paper, we develop a model called *TLDFP* to effectively predict minority disk failure leveraging the transfer learning, where traditional ML approaches perform poorly. Our main contributions include: **(1)** we are the first to define minority disk datasets and quantitatively evaluate them through extensive data analysis and experiments in data centers of heterogeneous disk models, **(2)** we are the first to present a novel method based on KLD values to select proper majority disk models, and **(3)** we develop a method of making crossing-disk model failure prediction, which has important practical applicability as different disk models are gradually placed into the realistic storage systems to replace failed disks. Our experiments with two datasets from real-world data centers have shown that *TLDFP* well outperforms representative traditional ML methods and existing transfer learning approaches in terms of failure detection rate and false alarm rate. *TLDFP* achieves on average 96% failure detection rate with only 0.5% false alarm rate in performing crossing-disk models failure prediction.

ACKNOWLEDGMENTS

This work is supported by the Innovation Group Project of the National Natural Science Foundation of China No.61821003, the National Key Research and Development Program of China under Grant No.2016YFB0800402 and the U.S. National Science Foundation grants CCF-1717660 and CCF-1813081. We owe my sincere gratitude to Yuanzhang Wang and Yangtao Wang who gave me their advice and time in listening to me. Thanks to Lihan Hu, Zhoubao sun and Zhenyuan Xu from Nanjing Audit University for their help. Ping Huang is the corresponding author of this paper.

REFERENCES

[1] Bruce Allen. 2004. Monitoring hard disks with SMART. *Linux Journal* 117 (2004), 60–65.

[2] Mirela Madalina Botezatu and Ioana Giurgiu et al. 2016. Predicting Disk Replacement towards Reliable Data Centers. In *Proceedings of the 22nd ACM SIGKDD, San Francisco, CA, USA, August 13-17*. 39–48.

[3] Brad Calder and Ju Wang et al. 2011. Windows Azure Storage: a highly available cloud storage service with strong consistency. In *Proceedings of the 23rd ACM SOSP, Cascais, Portugal, October 23-26*. 143–157.

[4] Corinna Cortes and Vladimir Vapnik. 1995. Support-Vector Networks. *Machine Learning* 20, 3 (1995), 273–297.

[5] Wenyuan Dai and Qiang Yang et al. 2009. Boosting for Transfer Learning. In *Proceedings of the 24th ICML*. 193–200.

[6] Brian Kulis et al. 2011. What You Saw is not What You Get: Domain Adaptation Using Asymmetric Kernel Transforms. In *Proceedings of the IEEE Conference on CVPR*. 1785–1792.

[7] Bianca Schroeder et al. 2016. Flash Reliability in Production: The Expected and the Unexpected. In *14th USENIX FAST, Santa Clara, CA, USA, February 22-25*. 67–80.

[8] Bingpeng Zhu et al. 2013. Proactive drive failure prediction for large scale storage systems. In *IEEE 29th Symposium on MSST, May 6-10, Long Beach, CA, USA*. 1–5.

[9] Cheng Huang et al. 2012. Erasure Coding in Windows Azure Storage. In *Proceedings of the USENIX ATC, Boston, MA, USA, June 13-15*. 15–26.

[10] Chang Xu et al. 2016. Health Status Assessment and Failure Prediction for Hard Drives with Recurrent Neural Networks. *TOC* 65, 11 (2016), 3502–3508.

[11] Eduardo Pinheiro et al. 2007. Failure Trends in a Large Disk Drive Population. In *5th USENIX FAST, February 13-16, San Jose, CA, USA*. 17–28.

[12] Farzaneh Mahdisoltani et al. 2017. Improving Storage System Reliability with Proactive Error Prediction. In *Proceedings of the USENIX ATC*. USENIX Association, Santa Clara, CA, 391–402.

[13] Haryadi S. Gunawi et al. 2018. Fail-Slow at Scale: Evidence of Hardware Performance Faults in Large Production Systems. *TOS* 14, 3 (2018), 23:1–23:26.

[14] Heng Wang et al. 2011. Action Recognition by Dense Trajectories. In *IEEE Conference on CVPR*. 3169—-3176.

[15] Joseph F Murray et al. 2003. Hard Drive Failure Prediction Using Non-Parametric Statistical Methods. In *Proceedings of the ICANN*. 1–4.

[16] Jing Li et al. 2014. Hard Drive Failure Prediction using Classification and Regression Trees. In *IEEE International Conference on DSN*. 383—-394.

[17] Jing Li et al. 2016. Being Accurate Is Not Enough: New Metrics for Disk Failure Prediction. In *35th IEEE SRDS, Budapest, Hungary, September 26-29*. 71–80.

[18] Jing Li et al. 2017. Hard drive failure prediction using Decision Trees. *Rel. Eng. & Sys. Safety* 164 (2017), 55–65.

[19] Justin Meza et al. 2015. A Large-Scale Study of Flash Memory Failures in the Field. In *Proceedings of the ACM SIGMETRICS, Portland, OR, USA, June 15-19*. 177–190.

[20] Joey Tianyi Zhou et al. 2014. Hybrid Heterogeneous Transfer Learning through Deep Learning. In *Proceedings of the 28 AAAI Conference on Artificial Intelligence, July 27 -31, Québec City, Québec, Canada*. 2213–2220.

[21] Lakshmi N. Bairavasundaram et al. 2007. An analysis of latent sector errors in disk drives. In *Proceedings of the ACM SIGMETRICS, San Diego, California, USA, June 12-16*. 289–300.

[22] Pereira et al. 2017. Transfer Learning for Bayesian Networks with Application on Hard Disk Drives Failure Prediction. In *Brazilian Conference on Intelligent Systems*. 228–233.

[23] Rong-En Fan et al. 2008. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research* 9 (2008), 1871–1874.

[24] Tomas Mikolov et al. 2009. Neural network based language models for highly inflective languages. In *Proceedings of the IEEE ICASSP, 19-24 April, Taipei, Taiwan*. 4725–4728.

[25] Tomas Mikolov et al. 2011. Extensions of recurrent neural network language model. In *Proceedings of the IEEE ICASSP, May 22-27, Prague Congress Center, Prague, Czech Republic*. 5528–5531.

[26] Teerat Pitakrat et al. 2013. A Comparison of Machine Learning Algorithms for Proactive Hard Disk Drive Failure Detection. In *Proceedings of the 13th International ACM ISARCS*. 17–21.

[27] Weihang Jiang et al. 2008. Are disks the dominant contributor for storage failures - A comprehensive study of storage subsystem failure characteristics. *TOS* 4, 3 (2008), 7:1–7:25.

[28] Wenjun Yang et al. 2015. Hard Drive Failure Prediction Using Big Data. In *IEEE 34th SRDS*. 13–18.

[29] Yong Xu et al. 2018. Improving Service Availability of Cloud Systems by Predicting Disk Error. In *USENIX ATC, Boston, MA, USA, July 11-13*. 481–494.

[30] Jerome H. Friedman. 2001. Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* 29, 5 (2001), 1189–1232.

[31] Sandipan et al. Ganguly. 2016. A Practical Approach to Hard Disk Failure Prediction in Cloud Platforms: Big Data Model for Failure Management in Datacenters. In *IEEE Second Big Data Service*. 105–116.

[32] Greg Hamerly and Charles Elkan. 2001. Bayesian Approaches to Failure Frediction for Disk Drives. In *Proceedings of the 18th ICML*. 202–209.

[33] Maayan Harel and Shie Mannor. 2011. Learning from Multiple Outlooks. In *Proceedings of the 28th ICML*. 401–408.

[34] Song Huang and Song Fu et al. 2015. Characterizing Disk Failures with Quantified Disk Degradation Signatures: An Early Experience. In *IEEE IISWC, Atlanta, GA, USA, October 4-6*. 150–159.

[35] Gordon Hughes and Joseph F Murray et al. 2002. Improved Disk-Drive Failure Warnings. *IEEE TOR* 51 (2002), 350—-357.

[36] Rie Johnson and Tong Zhang. 2014. Learning Nonlinear Functions Using Regularized Greedy Forest. *IEEE Trans. Pattern Anal. Mach. Intell.* 36, 5 (2014), 942–954.

[37] Ron Kohavi. 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *IJCAI*. 1137–1143.

[38] S. Kullback and R. A. Leibler. 79–86. On Information and Sufficiency. *The Annals of Mathematical Statistics* 22 (79–86), 1951.

[39] Joseph F. Murray and Gordon F. Hughes et al. 2005. Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application. *Journal of Machine Learning Research* 6 (2005), 783–816.

[40] Sinno Pan and Qiang Yang. 2009. A Survey on Transfer Learning. *IEEE TKDE* 22 (2009), 1345–1359.

[41] David A. Patterson and Garth A. Gibson et al. 1988. A Case for Redundant Arrays of Inexpensive Disks (RAID). In *Proceedings of the 1988 ACM SIGMOD, Chicago, Illinois, USA, June 1-3*. 109–116.

[42] David Powers. 2007. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies* 2 (01 2007), 37–63.

[43] Peter Prettenhofer and Benno Stein. 2010. Cross-language Text Classification Using Structural Correspondence Learning. In *Proceedings of the 48th ACL*. 1118–1127.

[44] Felix Salfner and Maren Lenk et al. 2010. A survey of online failure prediction methods. *ACM Comput. Surv.* 42, 3 (2010), 10:1–10:42.

[45] Kanoksri Sarinnapakorn and Miroslav Kubat. 2007. Combining Subclassifiers in Text Categorization: A DST-Based Solution and a Case Study. *IEEE TKDE* 19 (2007), 1638–1651.

[46] Bianca Schroeder and Garth A. Gibson. 2007. Disk Failures in the Real World: What Does an MTTF of 1, 000, 000 Hours Mean to You?. In *5th USENIX FAST, February 13-16, San Jose, CA, USA*. 1–16.

[47] H. Shimodaira. 2000. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference* 90, 2 (2000), 227–244.

[48] Igor V. Tetko and David J. Livingstone et al. 1995. Neural network studies, 1. Comparison of overfitting and overtraining. *Journal of Chemical Information and Computer Sciences* 35, 5 (1995), 826–833.