

HDDse: Enabling High-Dimensional Disk State Embedding for Generic Failure Detection System of Heterogeneous Disks in Large Data Centers

Ji Zhang, *Huazhong University of Science and Technology and University of Amsterdam;* Ping Huang, *Huazhong University of Science and Technology and Temple University;* Ke Zhou, *Huazhong University of Science and Technology;* Ming Xie, *Tencent Inc.;* Sebastian Schelter, *University of Amsterdam*

https://www.usenix.org/conference/atc20/presentation/zhang-ji

This paper is included in the Proceedings of the 2020 USENIX Annual Technical Conference.

July 15–17, 2020

978-1-939133-14-4

Open access to the Proceedings of the 2020 USENIX Annual Technical Conference is sponsored by USENIX.

HDDse: Enabling High-Dimensional Disk State Embedding for Generic Failure Detection System of Heterogeneous Disks in Large Data Centers

Ji Zhang[§†], Ping Huang[§£], Ke Zhou[§*], Ming Xie[ζ], Sebastian Schelter[†]
[§]*Huazhong University of Science and Technology*
[£]*Temple University,* [ζ]*Tencent Inc.,* [†]*University of Amsterdam*
Ji Zhang and Ping Huang are the co-first authors

Abstract

The reliability of a storage system is crucial in large data centers. Hard disks are widely used as primary storage devices in modern data centers, where disk failures constantly happen. Disk failures could lead to a serious system interrupt or even permanent data loss. Many hard disk failure detection approaches have been proposed to solve this problem. However, existing approaches are not generic models for heterogeneous disks in large data centers, e.g, most of the approaches only consider datasets consisting of disks from the same manufacturer (and often of the same disk models). Moreover, some approaches achieve high detection performance in most cases but can not deliver satisfactory results when the datasets of a relatively small amount of disks or have new datasets which have not been seen during training. In this paper, we propose a novel generic disk failure detection approach for heterogeneous disks that can not only deliver a better detective performance but also have good detective adaptability to the disks which have not appeared in training, even when dealing with imbalanced or a relatively small amount of disk datasets. We employ a Long Short-Term Memory (LSTM) based siamese network that can learn the dynamically changed long-term behavior of disk healthy statues. Moreover, this structure can generate a unified and efficient high dimensional disk state embeddings for failure detection of heterogeneous disks. Our evaluation results on two real-world data centers confirm that the proposed system is effective and outperforms several state-of-the-art approaches. Furthermore, we have successfully applied the proposed system to improve the reliability of a data center and exhibit practical long-term availability.

1 Introduction

Device failure is a common problem in large data centers, where hard disks are widely used as the primary storage devices. A disk failure could lead to temporary data loss and thus system breakdown, or even permanent data loss if the lost data cannot be recovered by data protection schemes (eg., replication and erasure codes) due to disk failures exceeding the designed correction capability [1]. About 80% of system breakdowns are caused by hard drive failures in the data center [2]. Therefore, how to ensure the reliability of disks becomes an important issue in a storage system [3]. Although there are a series of passive fault defense mechanisms like EC (Erasure Codes) [4, 5] and RAID (Redundant Arrays of Independent Disks) [6], many researchers have focused on proactive disk failure detection which aims to ensure the reliability and availability of large-scale storage systems in advance [7–9]. Disk failure prediction is an important issue in system researches. Timely and accurate failure prediction can ensure the operational continuity of systems and even avoid data loss. It is because of this common realization among the storage research community, there have recently emerged a decent amount of researches on disk failure prediction.

We observed six classes of approaches of disk failure detection: threshold-based (TB) approaches [10, 11], distance-based anomaly detection (DAD) approaches [12–16], shallow machine learning (SML)-based approaches [17–29], deep neural network (DNN)-based approaches [30–34], one-class classification (OCC) based approaches [35–39] and transfer learning (TL)-based approaches [1, 40–44]. However, these approaches have their limitations as summarized in Table 1.

(1) Limited applicability. Different disk manufacturers have different S.M.A.R.T (Self-Monitoring, Analysis, and Reporting Technology) values or data distribution, even in different disk models of the same manufacturer [1, 43]. This will result in a situation where methods (SML, DNN, OCC, TL) trained on specific parameters only work well for the same manufacturer, and often even for the same models, limiting their scope of applicability in practice.

(2) Lack of adaptability. New disk models enter gradually to replace or augment the storage capacity, leading storage systems to consist of disks from different vendors and/or different models [1, 41, 42]. Facing these different types of heterogeneous disks, the existing prediction models (DAD, SML, DNN, OCC, TL) must be retrained to obtain more reliable predictions in order to adapt to the changes in the data distribution introduced by these new disk models. Such re-

Table 1: Characteristics of different approaches in disk failure detection. ($\star$) refers to certain conditions that are required, e.g., finding a suitable source domain (i.e., another disk model) for knowledge transfer. All existing methods have certain disadvantages. We use TPR refers to True Positive Rate, FPR refers to False Positive Rate and F-Measure to evaluate these methods (for these evaluation metrics see Section 5.1.3).

	TB	DAD	SML	DNN	OCC	TL	HDDse
Applicability	√	√	×	×	×	×	√
Adaptability	√	√	×	×	×	√($\star$)	√
Imbalance datasets	√	×	×	×	√	×	√
Minority Disk	√	×	×	×	×	√($\star$)	√
Performance	TPR: 3%-10%	56%-70%	75%-96%	87%-98%	70%-92%	80%-97%	92%-97%
	FPR: 0%-2%	0%-1%	1%-4%	0%-1%	0%-10%	FPR: 0%-6%	0.2%-0.4%
	F-Measure: 2%-13%	49%-58%	67%-92%	86%-93%	65%-91%	77%-93%	91%-97%

training is tedious and expensive in a large data center.

(3) Imbalanced datasets. Imbalanced data refers to a situation where the number of samples is not the same for all the classes in a classification dataset. Most machine learning (ML) models tend to bias the class with the largest proportion of observations (known as majority class), which may lead to inaccuracies. This may be particularly problematic when we are interested in the correct classification of a "rare" class (also known as minority class). In real world cloud storage systems, the imbalanced ratio of positive (failure) samples and negative (healthy) samples poses a significant threat to the efficiency of machine learning models [2]. The most commonly used technique of existing methods (DAD, SML, DNN, TL) to approach this problem are under-sampling [1, 15, 28] and over-sampling techniques [45]. In under-sampling, the data samples are adjusted based on the class with the lowest sample count to keep the number of samples per class equal. In over-sampling the samples of the sparsely populated class are re-sampled to match the number of data samples in the other classes. Under-sampling discards a large amount of data while oversampling can easily cause overfitting in the model. Moreover, these approaches might increase the training cost.

(4) Minority disk failure detection. In large-scale storage systems, new disks gradually replace failed disks, which results in a situation where the data centers continuously contain small amounts of new disk models. Due to a lack of sufficient training data, some detective approaches (DAD, SML, DNN, OCC) fail to deliver satisfactory detective performance for these models. Although some TL methods (transfer learning is good at transferring knowledge from the source dataset to the target dataset) have been proposed to address this issue, their performance depends on finding a suitable source domain (in the form of another disk model) for knowledge transfer, which might be difficult in a real world data center (detailed in Section 3.1.2).

(5) Detective performance. The True Positive Rate (TPR), False Positive Rate (FPR) and F-Measure (all these evaluation metrics see Section 5.1.3 for details) are the commonly used metrics to measure the capabilities of classification models in disk failure detection [1, 15, 17, 19]. Most methods show unstable performance in practical long-term use and often fail to obtain both high TPR and low FPR.

In this paper, we propose a novel disk failure detection system called "High-Dimensional Disk State Embedding for Generic Failure Detection" (*HDDse*). It is a distance-based anomaly detection approach using deep learning and addresses the discussed shortcomings of the existing methods.

In summary, we provide the following contributions:

- To the best of our knowledge, we propose the first generic disk failure detection system *HDDse* for heterogeneous disks. It is not sensitive to imbalanced datasets, has wider applicability, well detects the minority disks and exhibits high adaptability. (Section 4.1)
- We propose a Long Short-Term Memory (LSTM)-based siamese network to calculate the similarity of disk health states in high-dimension space, which combines distance-based anomaly detection and deep learning to classify disk state in high-dimension. (Section 4.2)
- We experimentally evaluate our method on datasets from two real world data centers. We demonstrate that *HDDse* can effectively detect disk failures in long-term availability that greatly improves the reliability and availability of the storage system and outperforms the state-of-the-art approaches in five adopted metrics (Section 5.2 and 5.3).

2 Related Work

Existing work mostly uses the S.M.A.R.T (Self-Monitoring, Analysis and Reporting Technology) data to build a disk failure detection model. Almost all hard disk drives and flash-based SSDs now support S.M.A.R.T, which monitors the internal attributes of individual drives.

Threshold-based Methods (TB). All disk manufacturers use a thresholding algorithm which triggers a failure alarm when any single S.M.A.R.T attribute exceeds a predefined value [10, 11]. However, this approach provides only an estimated TPR (True Positive Rate, see Section 5.1.3 for details) of 3%-10% with a 0.1% FPR (False Positive Rate, see Section 5.1.3 for details). The reason is that the hard disk manufacturers set the thresholds conservatively to avoid expensive false alarm costs, i.e., they keep the FPR to a minimum at the expense of the TPR.

Distance-based Anomaly Detection Methods (DAD). DAD is the method of finding data objects with behaviors that

are very different from expectation based on some similarity metrics. Shen et al. [15] utilize the change in the Euclidean distance [46] between the healthy disk samples and the last sample of a failed drive. Wang et al. [13] propose a model for drive anomaly prediction based on Mahalanobis distance (MD). In their subsequent study [14], they use a generalized likelihood ratio test over the dissimilarity vector to detect disk failures. Huang et al. [16] explore the read/write head failures and bad sector failures using the Euclidean distance to calculate the dissimilarity between each S.M.A.R.T record prior to the failure and the failure record. Gao et al. [12] present an incremental detection model of disk failures based on the Euclidean distance to measure the local anomalies of test points within their isolation regions optimizing the judgment of test point anomalies.

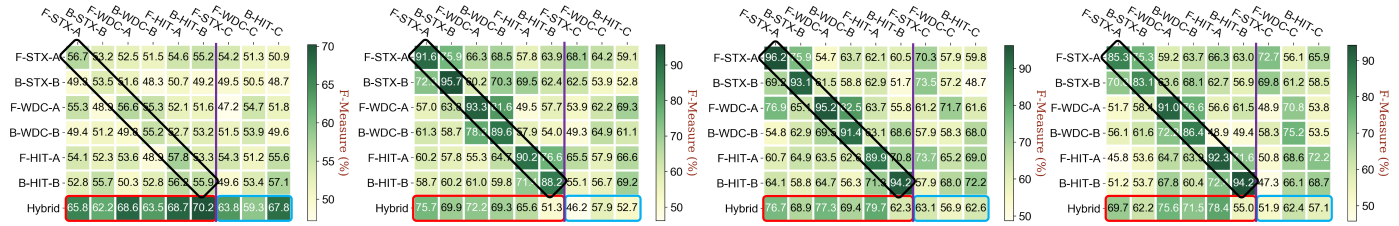
Shallow Machine Learning based Methods (SML). Conventional (shallow) machine learning methods typically require manual extraction and selection of features (eg., Support Vector Machines (SVM [47]), Logistic Regression (LR [48])), a critical step that is dispensed within the deep learning approach, i.e., it is automatic in deep learning approaches. Pitakrat et al. [29] evaluate and compare the performance of 21 machine learning algorithms for proactive disk failure detection. Li et al. [28] propose new drive failure detection models based on classification and regression trees. Yang et al. [27] design a disk failure prediction model based on L1-regularized logistic regression. Ganguly et al. [26] propose a two-stage ensemble predictive model. Chaves et al. [25] design a failure prediction method using a Bayesian Network. Mahdisoltani et al. [23] explore a range of different machine learning techniques (Classification and Regression Trees (CART [49]), Random Forests (RF [50]), SVM, Neural Networks (NN [51]) and LR) and show that sector errors can be predicted ahead of time with high accuracy using RF. Carlos et .al [22] investigate three different machine learning models (Decision Trees, NN, and LR). Xiao et al. [21] introduce a novel disk failure prediction model using Online Random Forests (ORFs). Aussel et al. [24] test several learning methods, SVM, RF and Gradient-Boosted Tree (GBT) and find that RF provides the best performance. Anantharaman et al. [20] experiment with two different types of ML approaches: RF and Long Short-Term Memory (LSTM) recurrent neural networks. Yong et al. [19] develop a cost-sensitive ranking-based ML model that can learn the characteristics of faulty disks from the past and rank the disks based on their error-proneness using the FastTree algorithm [52]. Yi et al. [18] predict the failure order with a LambdaMART [53] model. Xie et al. [17] propose an explanation approach designed for disk failure prediction.

Deep Neural Network based Methods (DNN). DNN method is an artificial neural network (ANN) with multiple layers between the input and output layers. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a nonlinear relationship. Xu et al. [34] introduce a method based on Recurrent Neural Networks (RNN) to assess the health statuses of disks. Pang [33] implement a Combined Bayesian Network (CBN) model to estimate the health degree of disks. Fernando et al. [32] present a remaining useful life estimation approach for disks by leveraging the capabilities of LSTM networks. Basak et al. [31] propose a data-driven framework based on LSTM for detection of whether a disk is going to fail in the next 7 days. Sun et al. [30] propose a temporal Convolutional Neural Network (CNN)-based model to predict failures.

One-Class Classification based Methods (OCC). OCC involves fitting a model on the normal data and predicting whether new data is normal or not. The imbalanced ratio of positive (failure) disk samples and negative (healthy) disk samples poses a significant challenge for efficient learning. Wang et al. [38, 39] present an anomaly detection method, which distinguishes anomalous behaviors from healthy events in each disk. Lucas et al. [37] propose a method for fault detection on hard disks that uses a Gaussian Mixture to model the behavior of only healthy hard disks. They also propose another method named GMFD [36], which applies a semi-parametric model (GMM) to build a statistical model using healthy disks. Francisco et al. [35] evaluate nine one-class classifiers for fault detection in hard disks.

Transfer Learning based Methods (TL). TL is an ML technique where a model trained on one task is re-purposed on a second related task. Transfer learning methods have been proposed to solve the minority disk failure problem. Botezatu et al. [43] apply a transfer learning approach to use a prediction model trained on a specific disk model for a new disk model of the same manufacturer. Francisco et al. [42] evaluate several transfer learning strategies on the task of failure prediction on hard disk drives. Their experiments state that transfer learning methods can provide performance gains in hard disk failure prediction models. Xie et al. [41] employ a simple but effective transfer learning method for disk failure detection which determines the best source from its anterior disk models and builds a transfer learning model with the help of the source domain. Zhang et al. [40,44] integrate both transfer learning and active learning techniques to detect the disk failure called ATAD and show its effective in cross-dataset time series anomaly detection. Zhang et al. [1] propose a minority disk failure prediction model named $TLDFP$ based on a transfer learning approach and use Kullback Leibler Divergence (KLD) as an effective indicator for source domain selection for large heterogeneous datasets.

Shortcomings of existing work. The above mentioned approaches (SML, DNN and OCC) deliver good detective performance only when both training and testing data are drawn from the same distribution [54]. A recent study [22] on heterogeneous disk failure prediction has pointed out that the predictive results are not good enough for adoption in prac-



(a) DAD approach [12] (b) SML approach [17] (c) DNN approach [31] (d) OCC approach [35]

Figure 1: Figure 1(a), 1(b), 1(c) and 1(d) respectively show the overall F-Measure (detailed in Section 5.1.3) of the four state-of-the-art methods using datasets of disk models from two data centers. Although the SML, DNN and OCC achieved better performance than the DAD method in the case of datasets consisting of disks from the same disk models, the DAD approach delivers strong applicability and adaptability.

tice. Therefore, most of the experiments for these methods only consider datasets consisting of disks from the same manufacturer (and often of the same disk models) and thereby have limited applicability. Except for the method DAD, the methods (SML, DNN and OCC) learn the S.M.A.R.T data distribution of the specific disk model but not the unified detection measure, as a result, these approaches have bad predictive performance when dealing with the disk models that have not yet appeared in previously trained models (i.e., poor adaptability). Furthermore, all the methods (especially for DNN and OCC) require a large number of training samples to build robust models, which is difficult to be satisfied for minority disks in data centers. Training models on the minority disks would dramatically increase the risk of overfitting, and the resulting poor generalization will decrease the performance of predictive models [1]. Although TL approaches can handle this situation well, an important premise of this approach is that there is one or more appropriate source majority disk models for knowledge transfer, but we find this to be a tough assumption in practice (see Section 3.1.2). Moreover, most of these approaches increase the training cost to process the imbalanced datasets and could result in discarding a large amount of data or model overfitting (mentioned in Section 1).

3 Preliminary Study and Motivation

In this section, we investigate the performance of existing approaches in three aspects (applicability, adaptability and minority disk failure detection) and describe our motivation for enabling high-dimensional disk state embedding for heterogeneous disk failure detection.

3.1 Preliminary Study

3.1.1 Applicability and Adaptability

We analyzed the overall performance of existing proposed approaches using different disk models from both the publicly available S.M.A.R.T dataset Backblaze[1] and the data center of Tencent (one of the largest social network companies in the world). Figure 1(a), 1(b), 1(c) and 1(d) respectively show the overall detective performance of the state-of-the-art methods (DAD [12], SML [17], DNN [31] and OC [35]) using different datasets of disk models. For convenience, we use `Data Center-Disk Manufacturers-Disk Model` to denote the dataset we use. For example, the disk model C from Seagate in data center BackBlaze can be referred to as `B-STX-C` and the disk model A from WDC in data center Tencent can be referred to as `F-WDC-A`.

In each figure, the vertical and horizontal axes refer to the disk models of training and testing dataset respectively. We use F-Measure (detailed in Section 5.1.3) to evaluate the performance of failure detection models. The higher the value is (the darker in the heatmap), the better the performance is. In the last row of each figure, we use the hybrid datasets (the first 6 disk models contain F-STX-A, B-STX-B, F-WDC-A, B-WDC-B, F-HIT-A and B-HIT-B) for training and then detect on different disk models. Note that the last three disk models F-STX-C, F-WDC-C, B-HIT-C are not included in the hybrid disk models for training. We summarize the findings from these four figures. **(1)** Detection approaches (SML, DNN and OCC) built on a specific disk model only has good results test on the same disk model (i.e., the detective model built on F-STX-A and detect on F-STX-A, as illustrated by the darker diagonal lines). Moreover, DNN-based approach shows the best performance in this case. **(2)** The SML, DNN and OCC approaches built on hybrid disk model datasets decrease the detective performance compared to a model built on the same disk model (we call the model has poor applicability in this case). Moreover, the performance of these approaches further deteriorates when detecting the disk models that have not appeared in the training datasets. **(3)** Although the overall detective results of the DAD method is not good enough to be deployed in practice, it is not sensitive to the different disk models. In other words, the performance of cross-model disk failure detection is very close. Besides, the DAD approach built on hybrid disks increases the performance compared to the model built on a single disk model (good applicability) and shows high adaptability to disk models that have not appeared in training. Note that we also evaluated many other studies on these methods and got similar results and we don't discuss all the results here due to space limitations.

3.1.2 Minority Disk Failure Detection

In order to investigate the detection for minority disks (the number of disks less than 1,500 [1]), for TL approach, we use the majority disk model which has the smallest Kullback Leibler Divergence (KLD) values with the detecting minority

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

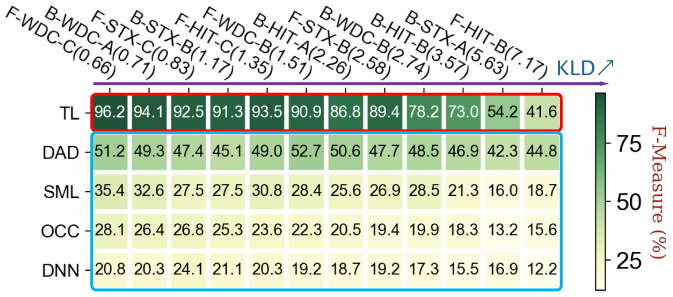


Figure 2: Minority disk failure detection using different approaches. The TL approach achieves the best performance but it depends on the small enough KLD value of the majority disk which is chosen for training.

disk model from the same manufacturer to train the detective models. The detecting minority disk models are listed on the x-axis ordered by the calculated smallest KLD values. For other approaches, they only trained the detective model based on minority disk datasets. Note that KLD is a metric measuring the divergence degree of one probability distribution from another expected probability distribution [55]. It indicates the disparities between two S.M.A.R.T datasets distributions. A zero KLD value means that the distributions of these two disk datasets are the same, while the KLD value increases as the differences between two data distributions widen. In general, the larger a KLD value is, the greater differences between two disk data distributions will be and the more difficult the knowledge transfer between two distributions will be in the transfer learning approach [1]. Figure 2 shows the results. The TL approach delivers the best detective performance on minority disk failure detection especially with the training datasets having smaller KLD values while other candidates are difficult to handle this situation.

Table 2: Distribution characteristics of the smallest KLD value for minority disk model in two data centers

Data Center	KLD(0∼1)	KLD(1∼2)	KLD(2∼3)	KLD(>3)
Tencent	**35%**	25%	23%	17%
Backblaze	**32%**	18%	31%	19%

In order to take an in-depth look at the feasibility of this method in practical large storage systems, we studied the disk quantities by the different KLD values in two data centers. As shown in Table 2, in Tencent data center, only 35% of all minority disk models could find a majority disk model with small KLD value (range from 0 to 1) for training. In other words, most minority disk models only find the majority disk models with a KLD value greater than 1 which might result in poor detection performance. A similar observation has been made in the data center Backblaze. Therefore, even in such large data centers with millions of disks, it is still difficult to find the most suitable majority disk model with small enough KLD value to use TL for minority disk failure detection.

3.2 Motivation

The preliminary study results above show that The DAD approach has better applicability and adaptability than other approaches (SML, DNN, OCC), and DNN approach gives the best detective performance. Before introducing our motivation, we first to answer the following three problems: **(1)** Why the DAD approaches have good applicability and high adaptability while DNN does not? The DAD method learns the distance (similarity) between the normal and abnormal disk samples in a certain space which has a commonality and not sensitive to the disk models. However, the DNN approach learns the distribution of S.M.A.R.T data which varies with disk models, the performance of this approach would decrease when the distribution changes. **(2)** Why the overall detective performance of the DAD method is not as good as other approaches. We think the reason is it performs distance-based transformation and computation in low-dimensional space but does not learn from the dynamically changed long-term behavior in high-dimension. **(3)** Why the DNN approach achieves the best performance among other candidates? DNN is good at mapping the raw low-dimensional S.M.A.R.T attributes to high-dimensional target spaces through complex transformations that perform good expression and fitting ability and thus achieves better performance.

Considering the above advantages of the DAD and DNN approaches, we are motivated to apply the distance-based anomaly detection approach and deep learning to build a general disk failure detection system for heterogeneous disks. In order to learn a unified measure of distance in high-dimension space using deep learning and then easily use the distance-based anomaly detection approach for comparison for two input data, we applied the siamese networks [56] (commonly used in the image recognition technology) which we will describe in detail in the next section.

4 Proposed System *HDDse*

In this section, we first provide an overview of our proposed system *HHDse* in Section 4.1. Then, we introduce the LSTM-based siamese network for disk failure detection in Section 4.2. Finally, we describe the sample pool and decision maker of *HDDse* respectively in Section 4.3 and Section 4.4.

4.1 System Overview

Figure 3 provides an overview of our proposed novel system *HDDse* which combines a sample pool, an LSTM-based siamese network for disk failure detection, and a decision maker. **Sample pool** (see Section 4.3 for details) is used to store the S.M.A.R.T instances collected daily for each disk. These instances are combined to form training and detecting samples for our approach. Note that the sample pool has both the disk S.M.A.R.T instances with ground truth (true labels with the disk states) for training and the unlabeled instances for detecting. **LSTM-based siamese network for disk failure detection** (see Section 4.2 for details) is the core part in *HDDse*, the input of the network is a pair of samples from the sample pool and the output is a binary classification result that indicates the similarity of these two samples. In online

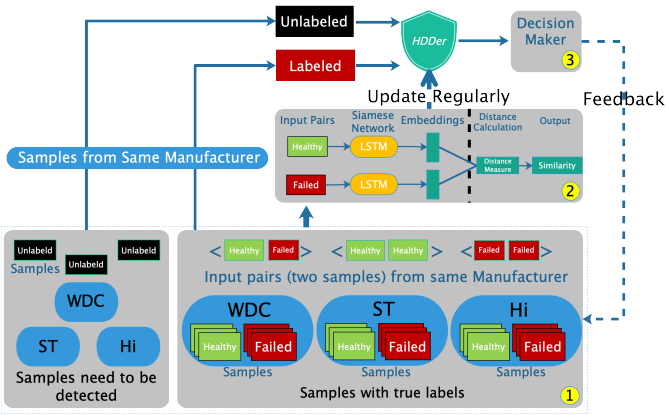


Figure 3: The overview of the proposed *HDDse*. It consists of a sample pool, an LSTM-based siamese network and a Decision Maker marked as 1, 2 and 3 respectively.

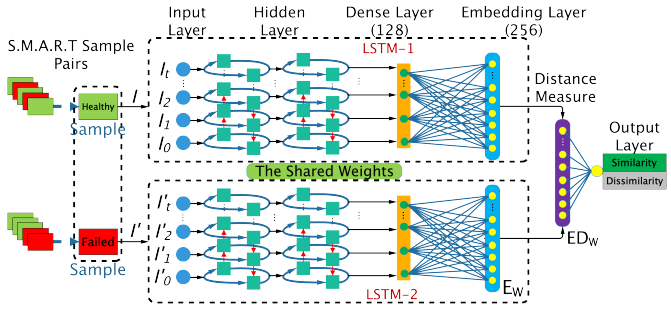


Figure 4: The structure of our proposed LSTM-based Siamese Network for Disk Failure Detection. The input is a pair of detecting samples and output is similarity.

detection, since there are many results of the target detecting samples at a continuous moment compared with other labeled samples, each output detective result indicates for a particular sample at different moments rather than the whole disk health state. Therefore, **Decision Maker** (see Section 4.4 for details) is a module to map these sub-results of samples to the final whole disk healthy state.

4.2 LSTM-based Siamese Network

Many previously proposed machine learning models, such as random forests (RFs), decision trees, SVM and DNN rely on the phenomenon that some key attributes are distinct from others when the disk is going to fail. Therefore, these approaches take a single snapshot of S.M.A.R.T attributes as training data for detection, without considering the sequential dependency between different statuses of a hard disk over time (only rely on features extracted from one day) because they are unable to make use of the time series data (except for converting them to sequential features manually). However, many researches have shown that the S.M.A.R.T of the disk changes dynamically with a certain trend [2, 19, 21, 27, 57], thus the current state of the disk may depend on a long-term historical trend. In contrast to all the aforementioned methods, we apply the LSTM to the parsing of sequential S.M.A.R.T information. LSTMs have been successfully applied to a variety of applications, including text sentiment classification [58] and multi-language text classification [59].

Traditional approaches to solving a classification problem, such as SVM, random forests (RFs) or even DNN, generally require that all the categories be known in advance (some newly disk models entered the data center without enough history information to classify the categories) for training. Moreover, those approaches are not suitable for applications where the number of samples per category is small (minority disk models). Siamese networks are dual-branch networks with tied shared weights, i.e., they consist of the same network copied. This method is proposed for training a similarity metric from data, which can be used for classification tasks (eg., face recognition) in which the categories need to be classified have not appeared during the training process, and can also handle the situation of training samples for a single category which is very small [60]. The symmetry of siamese networks is important and reasonable for learning a unified measure of distance, for example, the distance from disk state S_1 to state S_2 should be equal to the distance from S_2 to S_1.

Therefore, we propose an LSTM-based siamese network for disk failure detection. Figure 4 shows the structure of the network. The training sample pairs we feed to the network are randomly selected from the sample pool which contains two S.M.A.R.T samples (see Section 4.3 for details) from the same disk manufacturer denoted as $< S, S' >$. We design two LSTM networks to receive these two S.M.A.R.T samples respectively which need to be compared for their similarity. Note that these two LSTM networks (BiLSTM [61] refers to bi-direction LSTM layers) shared their weights in order to map the inputs to the same target high-dimension space for comparison. Each LSTM consists of an input layer U, four hidden layers H, one dense (fully connected) layer D with 128-dimensional units and an embedding layer E. In contrast to traditional neural networks, the LSTM operates over sequences of input vectors. This structure is able to capture the historical context of disk healthy statuses and makes LSTMs suitable for tasks related to sequential detection. Note that the dense layer is followed by an embedding layer (a fully connected layer) with 256-dimensional units, and then one more layer computing the distance metric (Euclidean distances [2]) between each siamese twin network. We employ a *Sigmoid* function in the final layer to output a normalized value on the learned high-dimensional feature space and scores the result between the feature vectors of the input sample pair. Note that our proposal is the first to propose an LSTM-based Siamese network based on the peculiar features of disk failures and has shown promising prediction outcomes.

Learning process. Let $E_\omega(S)$ and $E_\omega(S')$ be the projections of the input pair S and S' in the embedding space computed by the embedding layer (high-dimension space) network function E_ω. The output layer is a single unit computed by the induced distance metric between each siamese twin. The detective

[2] It is interesting to explore other sophisticated distance metrics to achieve better results.

vector is given by:

$$ED_\omega(S,S') = f_{sig}(\sum_{i=0}^{N-1} \beta_i |E^i_\omega(S) - E^i_\omega(S')|))$$

where f_{sig} is a *Sigmoid* activation function. β_i are additional parameters that are learned by the model during training, weighting the importance of the component-wise distance. Supposing that N represents the total number of S.M.A.R.T sample pairs over a dataset $D = < S_i, S'_i, Y_i >$, where i indexes the ith training pair and $Y(S^i, S'^i)$ is the corresponding label. We assume $Y(S_i, S'_i) = 1$ whenever S^i and S'_i are the same disk state label and $Y(S_i, S'_i) = 0$ otherwise. The total loss function is given by:

$$L_\omega(D) = \lambda||\omega||_2 + \frac{1}{2N}\sum_{i=0}^{N-1} \ell^i_\omega(S_i, S'_i, Y_i)$$

We use a squared L2-norm regularization (also called ridge regression) in this loss function to improve the ability of model generalization. The ω are the weights of the neural network and λ is the weight decay (this prevents the weights from growing too large and can be seen as gradient descent on a quadratic regularization term to prevent from the model overfitting) set as 0.001 to train our model. The instance loss function ℓ^i_ω comprises of terms including the similar (Y = 1) case (L_s), and the dissimilar (Y = 0) case (L_d):

$$\ell^i_\omega = Y_i \ell_s(S_i, S'_i) + (1 - Y_i)\ell_d(S_i, S'_i)$$

The loss functions for the similar and dissimilar cases are given by:

$$\ell_s(S,S') = (ED_\omega)^2$$

$$\ell_d(S,S') = \begin{cases} (m - ED_\omega)^2, & m < ED_\omega, \\ 0, & otherwise. \end{cases}$$

m is a margin which defines how far away the dissimilarities should be. These settings were chosen during cross validation, grid searching over possible margin settings. Suitable margin helps us distinguish the two input S.M.A.R.T samples better. Therefore, the instance loss function can also be given by:

$$\ell^i_\omega = Y_i(ED_\omega)^2 + (1 - Y_i)\{max(m - ED_\omega, 0)\}^2$$

It is interesting to investigate the proposed loss function. When the two samples are similar/dissimilar (Y = 1/ Y=0), if the ED_ω is wrongly calculated large/small by our model, the value of this loss function will become larger. Our goal is to minimize it.

The parameters of our model are optimized using the Adam optimizer [62] with a decreased decaying learning rate. The training process was run for 150,000 epochs with learning rate (lr) starting at 0.1 and decrease it by a factor of 2 every 50 training epochs. We use the dropout technique [63] (a dropout of 0.5) used on the recurrent units and between layers to prevent overfitting. For the hyper parameter (margin m, weight decay λ, learning rate, the unit number of fully-connected layers) optimization, we use the grid searching to perform hyper parameter selection.

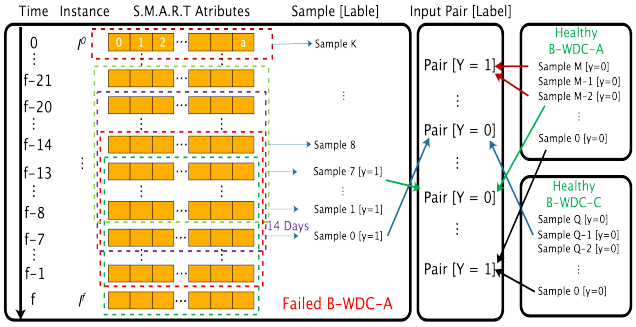


Figure 5: The relationship between S.M.A.R.T instances, samples and the input pairs in our approach. Each sample consists of 14 continuous instances and the continuous samples in a period of seven days before actual failures are labeled as failed.

4.3 Sample Pool for Imbalanced Datasets

Our sample pool consists of the collected S.M.A.R.T instances with their corresponding confirmed labels and some instances that need to be detected. We collect all the instances of each disk in the data center every day. All the time data have been discretely sampled at an interval of one day. If the data collection starts at day T_0 and a disk can fail in any day T_f after that, a disk can have data for all such time indices t where t varies from T_0 to T_f. We can formulate this instance as a multivariate sequential vector $I^t = \{I^t_0, I^t_1, I^t_2, ..., I^t_a\}$ that contain length-a attributes of S.M.A.R.T data at time t for each disk. More specific information about the attributes we use in our evaluation is given in Table 3. Besides, each input of the LSTM consists of a fixed length (we set 14 days in our system to learn the long-term disk state behavior) continuous instances (referred to one sample) with its corresponding label. The first day recorded in a disk failed sample is not the day when the disk fails, but the latest day when the collected attribute stops changing [2, 21, 23]. We use change-point detection to decide how many days samples of failed disks preserve for training. Change-point detection can find an abrupt change during a period. We observe that most attributes have a significant change around 7 days before failures. Therefore, for a failed disk, continuous samples in a period of 7 days (our method has the ability to predict the failure 1-7 days in advance) before actual failure (T_f) are labeled as failed ($y = 1$). For healthy disks, we label the total continuous samples as healthy ($y = 0$). Note that Y = 1 (Y = 0) means the label of the generated input pairs which is a combination of two same (different) labeled samples. The relationship between S.M.A.R.T instances, samples and the input training pairs is shown in the Figure 5. RAID arrays will not affect the operations in the sample pool and the decision maker in Section 4.4.

As mentioned in Section 4.2, the inputs of our proposed network are two S.M.A.R.T samples $< S, S' >$ in pairs. There-

fore, we can freely combine and label a pair of samples from the same disk manufacturer. There are two benefits for generating this form of training datasets compared to existing approaches which treat each S.M.A.R.T sample as a snapshot in the training process.

Better with imbalanced datasets. It reduces the degree of data imbalance by the simplest free combination. We use the imbalance degree (*IDe*) [64] to indicate the dataset imbalance which is defined as the ratio between the majority and minority samples (the larger value the *IDe* is when *IDe* is larger than 1, the more imbalanced the dataset is). For an imbalanced dataset containing a minority class sample with size A and the *IDe* is α, the majority class sample size is αA. The number of pairs with labels $Y = 1$ ($Y = 0$) n_1 (n_0) after combining the input samples can be expressed as: $\begin{cases} n_1 = C_A^2 + C_{\alpha A}^2, \\ n_0 = C_A^1 \times C_{\alpha A}^1. \end{cases}$ where the C_A^2 ($C_{\alpha A}^2$) is the number of "similar" sample pairs generated by the minority class samples (majority class samples). The new imbalance degree IDe' of the input pairs is given by: $IDe' = \frac{n_1}{n_0} = \frac{\alpha}{2} - \frac{1+\alpha-A}{2A\alpha}$ since $A, \alpha > 1$, the value of IDe' will be around $\frac{\alpha}{2}$, which effectively alleviates the original data imbalance by a factor of two. Note that we directly arrange all labeled samples to form the input training pairs without losing large amounts of information (instances) in the sample pool compared to the Under-sampling method which is commonly used in recent researches. Although our method can alleviate imbalanced datasets, it is difficult to eliminate. Fortunately, extensive experimental results in Section 5 demonstrate that the siamese network is insensitive to imbalanced datasets, which is the same as many existing research results [65, 66].

Better with minority disk models. It forms large training samples even with the minority disk models. The number of training pairs with the minority disk models in existing methods is $P = A(1+\alpha)$. However, in our method the number of training pairs is: $\frac{P!}{2!(P-2)!} = \frac{P(P-1)}{2}$ which is extremely large compared to existing methods. Therefore, our model can increases the number of samples greatly and make better use of deep learning algorithms to detect the failure and avoid model overfitting (the experimental results are discussed in Section 5.2.2).

4.4 Decision Maker in *HDDse*

In this section, we aim to seek answers to the following two problems: **(1) Which** training samples need to be arranged to form the input pairs with the detecting sample and how to make a decision (same state or not) for these pairs in the online detection process. **(2) How** to make a disk healthy state decision for a target disk based on the results of these detecting samples at continuous moments?

When a sample of a target detecting disk needs to be detected, we let all the labeled training samples from the same disk manufacturer to be arranged to form the input pairs. For accelerating the process of making a decision for each sample, we follow four detecting steps:

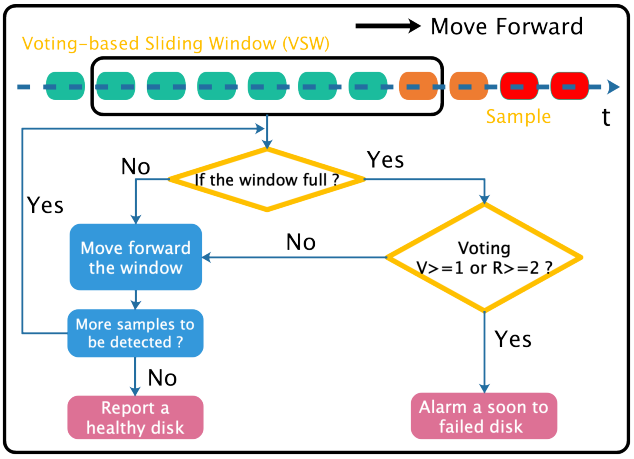


Figure 6: The flowchart of the voting-based sliding window. Moving forward until the disk health statue is reported.

Step 1. We arrange all the samples labeled as failed ($y = 1$) from the same disk model (all majority and minority disks) to form the input pairs with the target detecting sample respectively. If any of these pairs are detected as similar ($Y = 1$), we will mark this detecting sample as failed ($y = 1$), otherwise, proceed to the next step.

Step 2. Considering the sparsity of the state of healthy disk samples, we first randomly select 10% healthy disks of the same disk model and arrange the samples ($y = 0$) collected randomly in one-month intervals to form the input pairs with the detecting sample. If all these pairs are detected as similar ($Y = 1$), we will mark this sample as healthy ($y = 0$). Otherwise, follow the third step.

Step 3. We compare with the samples labeled as failed ($y = 1$) from the different disk models but the same manufacturer with the detecting sample. If more than half of the results are detected as similar ($Y = 1$), we will mark this sample as failed ($y = 1$).

Step 4. Similar to step 2, the only difference is we arrange the different disk models from the same manufacturer. Note that when the disk model of the detecting sample did not exist in the sample pool, we will start from step 3.

Each target detecting disk consists of several results of detecting samples at continuous moments, each output result indicates the detection for a particular sample at one moment rather than the whole disk health state in a period. Therefore, to improve the robustness of the detection method against noise, we propose a voting-based sliding window (*VSW*) method to make a disk healthy state decision for the final disk state. Figure 6 shows the architecture of *VSW*. According to the results of the detecting samples stored in the Decision Maker, we define the *VSW* method for failure detection in the following manner: Define a length-W time sliding window and move it forward everyday. A failure alarm will be reported, if the window consists of V (R) consecutive results from step 1 (step 3), otherwise, there is no failure alarm. These parameters (W, V, R) will be optimized as hyper parameters of the model to determine its optimal value and were chosen

during cross validation, grid searching over possible settings. The configuration of our real-world large scale storage system is $W = 7, V = 1, R = 2$. Note that some other decision solutions can be explored to improve the robustness of our method further (eg., a weighted k-nearest neighbors approach).

5 Experimental Evaluation

In this section, we evaluate the detective performance of *HDDse*. We first describe the methodology, followed by the experimental results of effectiveness and efficiency and compare *HDDse* against the state-of-the-art approaches with respect to the evaluation metrics.

5.1 Methodology

We describe the characteristics of two real world S.M.A.R.T datasets and the attributes selection in our experiments. Then we introduce the experiment setup and some evaluation metrics used in our experiments.

5.1.1 Datasets and Attribute Selections.

Datasets. We use S.M.A.R.T datasets from two real world data centers for evaluation. One is the publicly available data set from "Backblaze"[3], which spans a period of 58 months consisting of 146,203 healthy disks and 8,256 failed disks. The second proprietary dataset has been collected by Tencent and spans 29 months consisting of 70,192 healthy disks and 2,971 failed disks. All disks in these datasets were labeled to be either failed or healthy. Note that the data from these two data centers are extremely imbalanced. We tried to impute the missing S.M.A.R.T values or disks by replacing them with the median value.

Attribute Selections. Each S.M.A.R.T instance contains many meaningful attributes. We first select all the common attributes of the disk manufacturers. However, we find some attributes are irrelevant to disk failure events because they are immutable or have not experienced noticeable abnormal changes. Therefore, we use correlation coefficients and select nine attributes that correlate most with disk failure. The selected attributes are listed in Table 3. Each SMART attribute entry consists of many elements. In our paper, we focus on the three elements $(ID, Normalized\,value, Raw\,value)$ in our collected datasets. Since different attributes have different output ranges, (which might lead to different impacts on the detection model), we normalize the range of all S.M.A.R.T attributes using min-max normalization, a common preprocessing technology in machine learning: $x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}}$ where x is the original value of a S.M.A.R.T attribute, x_{max} and x_{min} are the maximum and minimum value of the attribute in the training data set, respectively. Note that we tried other normalization methods (e.g., z-score), but achieved the best results using min-max normalization. The hyper parameters and attribute selections were done over the entire datasets.

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

Table 3: The S.M.A.R.T Attributes for Our Evaluations

#ID	S.M.A.R.T Attribute Name	Attribute type
001	Raw Read Error Rate	Normalized
004	Start/Stop Count	Raw
005	Reallocated Sectors Count	Raw
012	Power Cycle Count	Raw
187	Reported Uncorrectable Errors	Normalized
193	Load Cycle Count	Normalized
196	Reallocation Event Count	Raw
197	Current Pending Sector Count	Raw
198	Offline Uncorrectable Sector Count	Raw

5.1.2 Experiment Setup.

To simulate the detecting process of disk failure in the real world, we use the following method to build the experimental datasets. All disks are randomly divided into a training set and a testing set at a ratio of 7 to 3, as in most researches [1, 35, 67, 68] which guarantees that the failed disks in the training set and the testing set are completely independent. Note that the deterioration of a failed disk is a gradual process from healthy to failure, and not all samples of failed disks need to be used in the training set; otherwise, those healthy samples of failed disks which are far from the actual failure would disturb the training of the detection model. Therefore, only the last seven continuous samples (we detailed it in Section 4.3) before the moment of failure of the training disks can be regarded as failed samples and need to be added to the training dataset. Furthermore, we obtain all results via cross-validation [69] to decrease the variability of the detections (analogous to many methods [1, 41, 70]). For the configurations and parameters of our system *HDDse*, the maximum number of training epochs is set to 1000; the learning rate is initially set to 0.1, and we decrease it by a factor of 2 every 50 training epochs; the coefficient of weight decay λ is set to 10^{-8}. We train and evaluate our method on a Linux server with 12-core 4.0GHz CPU, 64GB RAM and 200GB HDD.

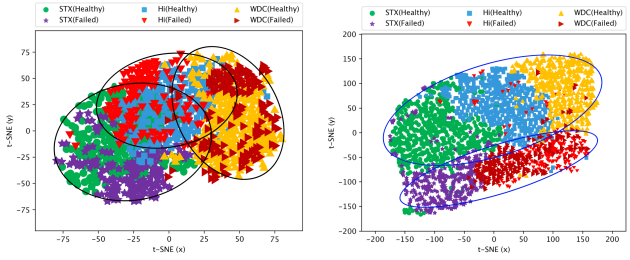
5.1.3 Evaluation Metrics.

We use the following five metrics to report the detective performance in our experiments which are commonly used for evaluating the capability of disk failure detection approaches.

TPR. True Positive Rate (also called recall) is the proportion of failed disks that are correctly predicted. The higher the TPR is, the better the model is.

FPR. False Positive Rate (also called false alarm rate) is the proportion of healthy disks that are falsely predicted as failed. The lower the FPR is, the better the model is.

AUC. We use the AUC (Area under the receiver operating characteristic curve) value under the ROC curve (receiver operating characteristic) to evaluate the binary classification performance of our detection model in imbalanced datasets. ROC is a curve plotting the TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis. Therefore, the larger the AUC value, the higher the TPR and the lower the FPR. AUC is the area under this curve. A higher AUC means the model is better at distinguishing failed and healthy disks.



(a) Before embedding. (b) After embedding.

Figure 7: The t-SNE of the S.M.A.R.T data before and after embedding using our approach. After the disk state embedding, the healthy and failed disks are nearly clustered together and easily separated.

F-Measure. F-Measure is a balance between the two metrics TPR and Prediction Precision (PP). PP is the proportion of detected failed disks that are correctly detected. The higher the F-Measure is, the better the model is.

C-MTTDL. We use Cost-based MTTDL (Mean Time To Data Loss) to evaluate the reliability and availability of the storage system. MTTDL [71] was proposed to approximate the mean time to data loss with failure detection model which is given by: $MTTDL \approx \frac{MTTF}{1-\frac{k\mu}{\mu+\gamma}}$ where $MTTF$ is the Mean Time To Failure of a disk. k is the TPR and γ is the inverse of how far in advance the model can detect the impending failures. μ is the inverse value of Mean Time To Repair (MTTR, replace a new disk or transfer the data of the failing disk to a new one). However, this metric only demonstrates the relationship between the MTTDL and the TPR (correctly detected) but neglects the cost of misclassification by the approach. Too many misclassifications will result in unavailability of the storage system. Therefore, we propose an end-to-end economic analysis metric called the Cost-based MTTDL ($C-MMTDL$) which considers not only the reliability but also the misclassifications cost (considering the false positive rate and cost of replacing disks). We will give the detailed definition of $C-MTTDL$ in Section 5.2.5. Although we propose the $C-MTTDL$, it is interesting to explore other end-to-end evaluation metrics (benefit in customer-perceived latency or throughput from accurate prediction).

5.2 Effectiveness Comparison

In this section, we first analyze the disk state embeddings learned by our LSTM-based Siamese Network. Then we show the results of *HDDse* compared to several state-of-the-art approaches in aspects of the ability of the minority disk detection, model applicability and adaptability respectively.

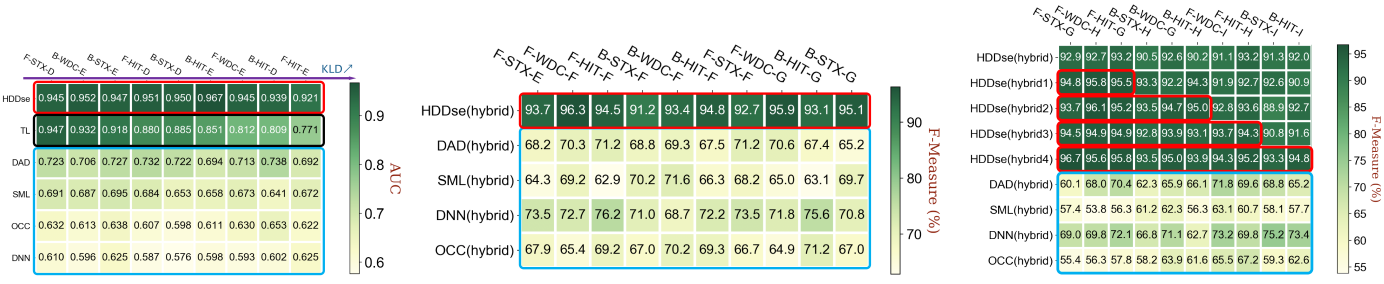
5.2.1 Analysis of Disk State Embeddings

To visualize the high-dimension embedded disk statues of the embedding layers in our approach (See Figure 4), we use the t-Distributed Stochastic Neighbor Embedding (t-SNE) [72] technique which can project high-dimensional embedding spaces into 2D spaces for visualization while striving to keep data clustered together in the high dimensional space clustered together in the low-dimensional space as well.

For comparison, Figure 7 shows the t-SNE of the S.M.A.R.T data attributes in low and high dimensionality respectively (before and after embedding using *HDDse*) based on the collected data from three disk manufacturers. In our case, the low dimensionality is the original S.M.A.R.T attributes most of the recently proposed methods (DAD, SML, DNN, OCC and TL) leveraged (the specific attributes are listed in Table 3). Note that the x and y axes of a particular point have no meaning on their own and the t-SNE only attempts to preserve clusters in higher dimensional space. The data points have been colored and shaped based on disk statues and their manufacturers. It is observed the data points of these three disk manufacturers cluster relatively closely and the relationship between the healthy and failed disks is hard to distinguish with a unified method shown in Figure 7(a). This highlights the challenges in failure detection of disks based on S.M.A.R.T attributes in heterogeneous populations. This result is consistent with the findings of the research [70] and results in most of the related works only considered population consisting of the same disk models in their experiments. As can be seen from Figure 7(b), the healthy and failed disks are easily separated (the healthy state is above the failed state for all the disks from different manufacturers) after the embedding using our proposed approach. This experiment demonstrates that our *HDDse* can generate a unified and efficient high-dimensional disk state embeddings for generic disk failure detection of heterogeneous disks.

5.2.2 *HHDse* only Trained on Minority Disk Datasets.

As mentioned in Section 4.3, the input training pairs generated by our method is extremely large compared to the original samples of existing approaches. It is interesting to investigate the detective performance of *HHDse* when only trained on minority disk datasets. We compare our approach with the other five state-of-the-art approaches: DAD [12], SML [17], OCC [35], DNN [31] and TL [1]. For a fair comparison, the TL approach uses the majority disk model which has the smallest KLD value with the detecting minority disk model from the same manufacturer to train the detective models. Minority disk models from different data centers and manufacturers are listed in the x-axis ordered by these calculated the smallest KLD values. For other approaches, they only trained the detective model based on minority disk datasets. As can be seen from Figure 8(a), none of the four approaches (DAD, SML, OCC and DNN) can deliver a high AUC value. The poor detective performance is due to overfitting caused by using small homogeneous datasets [1]. In particular, DNN achieved the worst results because the neural network required a huge number of samples to fit a large number of weights. Besides, the results of the TL method imply it largely depends on whether you can find the smallest KLD value majority disk training dataset for modeling as discussed in Section 3.1.2. It is worthy to note that our *HDDse* achieves the best performance in all cases. The reason is two-fold. On the one hand, the training pairs generated by our method are extremely large which makes our model not easy to get overfitting. On the other



(a) (b) (c)

Figure 8: (a) Performance comparison for minority disk failure detection of different approaches. *HDDse* delivers the highest AUC in most cases. (b) Performance comparison for disk failure detection using hybrid disk datasets. *HDDse* achieves the best detection results in all cases which shows good applicability. (c) Performance comparison for disk failure detection using different hybrid disk datasets. *HDDse* achieves the best detection results even the disk models have not appeared for training which shows good adaptability.

hand, the disk state embeddings in high-dimension learned by our designed LSTM-based Siamese Network is effective for model classification.

5.2.3 The Applicability of *HHDse*

In order to investigate the model applicability of our *HDDse*, we use the imbalanced datasets of ten disk models from three disk manufacturers in two data centers to conduct this experiment. We compare our approach with the other four approaches: DAD, SML, OCC and DNN. Considering the generality, those ten detecting disk models consist of three from the data center Tencent (F-STX-E, F-WDC-F and F-Hi-F), three from data center BackBlaze (B-STX-F, B-WDC-F, B-HIT-F) and four minority disk models from two data centers (F-STX-F, F-WDC-G, B-HIT-G, B-STX-G). We use all these hybrid disk datasets to train and detect these models respectively, and the detective results are shown in Figure 8(b). When dealing with the disks from different manufacturers, data centers and even minority disk models, *HDDse* achieves higher F-Measure values compared to other candidates. The main reason is that our model maps low-dimensional attributes (disk status) to a general high-dimensional space that is not sensitive to the differences in disk models, which leads to the detection to be performed well using a unified distance calculation. This experiment demonstrates that *HDDse* has good applicability for disk failure detection in a heterogeneous environment.

5.2.4 The Adaptability of *HHDse*

We evaluate the adaptability of our *HHDse*, e.g., how an approach can adapt to a disk model that has not appeared in the training dataset. We use another ten disk models (note that they are different from the ten disk models showed in Figure 8(b)) from three disk manufacturers in two data centers to conduct this experiment. As can be seen from Figure 8(c). Data set "Hybrid" contains the same ten disk models for training described in Figure 8(b). "Hybrid1" contains another three disk models from data center Tencent (F-STX-G, F-WDC-H and F-Hi-G) compared to "Hybrid", dataset "Hybrid2" add another three from data center BackBlaze (B-STX-H, B-WDC-G, B-Hi-H), "Hybrid3" has two more minority disk models from data center Tencent (F-WDC-I , F-HIT-H) than "Hybrid2" and "Hybrid4" includes all 10 disk models in the x-axis compared to the dataset "Hybrid". For a fair comparison, we use the dataset "Hybrid" to train the DAD, SML, DNN, OCC approaches. The results are shown in Figure 8(c). Our method using different hybrid datasets all achieved better detective performance than other candidates. It is worthy to note that the datasets of the detecting disk models have not appeared for training delivers comparable performance as those disk models contained (framed by red lines) in training data sets in the first five rows of Figure 8(c). It indicates that our *HDDse* does not need to establish or maintain a new model and owns strong adaptability that can completely adapt to a new disk model regardless of disk manufacturers, disk models, data centers, and minority disks. Note that we also verified the HDDse's performance on NVMe SSD and obtained promising results, which are not included due to the space limit.

5.2.5 Improvement of Storage System Reliability

As the above comprehensive analysis of detective results described, we achieved better AUC and F-Measure compared to other approaches. In this section, we will first give the definition of the economic analysis metric $C-MTTDL$ and then quantitatively evaluate the reliability and availability of the system based on different approaches. $C-MTTDL$ can be expressed as: $C-MTTDL = \frac{MTTDL}{Cost} \approx \frac{MTTF}{(1-\frac{k\mu}{\mu+\gamma})(C_aFP+C_bFN)}$ where MTTDL was defined in Section 5.1.3 and FP (FN) is the number of true healthy (failed) disks that are falsely predicted as failed (healthy). C_a and C_b are the corresponding cost of these misclassifications. These two factors can be set differently and obtain different results according to the model maintenance requirement (cost sensitivity) in the real world. C_a and C_b are set to 200 and 100 (dollars) respectively in our evaluation which were estimated in Tencent data center. $C-MTTDL$ denotes the MTTDL per dollar cost and the larger the better. The larger the $C-MMTDL$ is, the better is the reliability and availability the storage. We investigate the $C-MTTDL$ in based on the datasets from data center Tencent. We assume all models can detect the impending failures seven days in advance ($\gamma = 1/(7*24hours)$, the inverse value of $MTTR$ $\mu = 1/10hours$ and $MTTF$ = 1,390,000 hours. All these parameters come from data center Tencent and our experiments. We list the results in Table 4, which shows our

approach improves the $C-MTTDL$ by about 2 orders of magnitudes than the other four candidates. In other words, in addition to ensuring the detective performance of our approach *HDDse*, we have greatly improved the reliability of the storage system at a lower cost.

Table 4: Improvement of C-MTTDL

Method	$k(TPR)$	FP	FN	Cost	MTTDL (years)	C-MTTDL (hours/dollar)
OCC	**62.6%**	8212	1062	1,748,600	397.6	1.94
DAD	**45.2%**	3422	1537	838,100	276.7	2.89
SML	**72.6%**	6159	783	1,310,100	504.10	3.37
DNN	**85.3%**	4791	419	1,000,100	814.13	7.13
HDDse	**95.8%**	103	140	34,600	1656.3	419.35

5.3 Efficiency Comparison

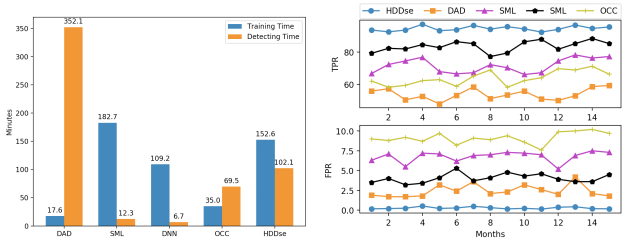
In this section, we evaluate the effect on training/detecting time and the practical long-term availability of our approach *HDDse* compared to the state-of-the-art approaches.

5.3.1 Training and Detecting Time

We train and evaluate all the approaches on a Linux server with 12-core 4.0GHz CPU, 64GB RAM and 200GB HDD. All the approaches record the total training (from start until convergence is achieved, i.e. little changes in the performance) and online detecting time (it is for the entire testing set). Figure 9(a) provides further time statistics for each approach. DAD has the lowest training time and the highest detecting due to its easy training method and large computation of calculating distances. Our approach *HDDse* takes the second-highest training time followed by the SML method (Random Forests) and the second-highest detecting time followed by the DAD method. This is attributed to our method generating large training pairs compared to the original datasets that need more time to converge. Moreover, it needs more time compared to other training samples to determine the disk statue in the detecting process (we use some sampling methods to accelerate the process of detecting mentioned in Section 4.4). Considering the training task in a data center and sometimes, the detective models are updated weekly or monthly (we perform once a week), so the time cost of *HDDse* is acceptable. Note that the time cost can be further shortened if GPU and many model compression and acceleration technologies [73] are used. We leave in our future work to explore other solutions for optimizing the efficiency of our approach.

5.3.2 Evaluating Practical Long-Term Availability

We last simulate practical long-term availability in data centers Tencent. As we mentioned in section 4.2, S.M.A.R.T data of the disk changes dynamically with a certain trend, thus the distribution of S.M.A.R.T attributes changes over time, resulting in unstable detective performance for many approaches. However, it is important to design a stable detection approach for large data centers without additional manual model tuning. To fairly evaluate the detective performance of different approaches in the long term, we employ the same accumulation strategy to update the model periodically e.g., once a week, using all the training data collected from the beginning. Figure 9(b) shows the TPRs and FPRs of the different detective approaches in the following 15 months. As can be seen, *HDDse* shows higher TPR and lower FPR compared to other state-of-the-art methods. It is worthy to note that *HDDse* exhibits stable detective performance than other candidates. We attribute this to the LSTM-based network which is well learned from the dynamically changed long-term behavior of disk statues. Similar to most of related works, we are focused on the predictive accuracy, because designing an accurate approach is the first critical step toward building a robust, highly reliable, and readily available operational storage system. With a high-accuracy failure prediction approach in place, we will have a high level of confidence in integrating it into the system, which is more of a mechanism rather than a policy. For instance, a direct application is to use the prediction results to perform data backups and replace disks that are about to fail to prevent data loss. Moreover, we can use the predictive results to analyze the mechanism of disk sector error and build a sector error predictive model to accelerate the scrubbing rate of disks to find the sector errors in advance and improve the reliability of the storage system.



(a) (b)

Figure 9: (a) Training and detecting time comparison for different approaches. (b) TPR and FPR of different approaches. *HDDse* delivers higher TPR and lower FPR in a stable manner which shows long-term availability.

6 Conclusion

Disk failures have become one prevailing reason for unexpected system unavailability. In this paper, we propose *HDDse*, an LSTM-based siamese network that can learn the dynamically changed long-term behavior of disk healthy statues and generate a unified and efficient high dimensional disk state embeddings from low dimensional S.M.A.R.T attributes for disk failure detection. We evaluate our approach using two real-world datasets to demonstrate that *HDDse* is effective and outperforms several state-of-the-art approaches. Specifically, *HDDse* has good detective adaptability to the disks which have not appeared in training and deliver good performance for the imbalance or minority disk datasets, thus improving storage system availability. Furthermore, the proposed approach improves the reliability of a data center and exhibits long-term availability.

Acknowledgments

We thank the anonymous reviewers and our shepherd Eno Thereska for their help in improving our paper. This work is supported by the Innovation Group Project of the National Natural Science Foundation of China No.61821003.

References

[1] Ji Zhang, Ke Zhou, Ping Huang, Xubin He, Zhili Xiao, Bin Cheng, Yongguang Ji, and Yinhu Wang. Transfer learning based failure prediction for minority disks in large data centers of heterogeneous disk systems. In *Proceedings of the 48th International Conference on Parallel Processing*, ICPP, pages 66:1–66:10. ACM, 2019.

[2] Xiaoyi Sun, Krishnendu Chakrabarty, Ruirui Huang, Yiquan Chen, Bing Zhao, Hai Cao, Yinhe Han, Xiaoyao Liang, and Li Jiang. System-level hardware failure prediction using deep learning. In *Proceedings of the 56th Annual Design Automation Conference 2019*, DAC '19, pages 20:1–20:6. ACM, 2019.

[3] Saurabh Kadekodi, K. V. Rashmi, and Gregory R. Ganger. Cluster storage systems gotta have heart: improving storage efficiency by exploiting disk-reliability heterogeneity. In *17th USENIX Conference on File and Storage Technologies (FAST 19)*, pages 345–358. USENIX Association, 2019.

[4] Cheng Huang, Huseyin Simitci, Yikang Xu, Aaron Ogus, Brad Calder, Parikshit Gopalan, Jin Li, and Sergey Yekhanin. Erasure coding in windows azure storage. In *Presented as part of the 2012 USENIX Annual Technical Conference (USENIX ATC)*, pages 15–26, 2012.

[5] Brad Calder, Ju Wang, Aaron Ogus, Niranjan Nilakantan, and Arild et. al Skjolsvold. Windows azure storage: A highly available cloud storage service with strong consistency. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, pages 143–157. ACM.

[6] David A. Patterson, Garth Gibson, and Randy H. Katz. A case for redundant arrays of inexpensive disks (raid). In *Proceedings of the 1988 ACM SIGMOD International Conference on Management of Data*, SIGMOD '88, pages 109–116. ACM.

[7] Jayanta Basak and Randy H. Katz. Significance of disk failure prediction in datacenters, 2017.

[8] Drew D. Penney and Lizhong Chen. A survey of machine learning applied to computer architecture design, 2019.

[9] Ao Ma, Fred Douglis, Guanlin Lu, Darren Sawyer, Surendar Chandra, and Windsor Hsu. Raidshield: Characterizing, monitoring, and proactively protecting against disk failures. In *13th USENIX Conference on File and Storage Technologies (FAST 15)*, pages 241–256. USENIX Association, 2015.

[10] Allen Bruce. Monitoring hard disks with smart. *Linux Journal*, 117, 2004.

[11] Richard Nass. Smart failure-prediction method now being endorsed for scsi disk drives. *Electronic Design*, 43, 1995.

[12] X. Gao, S. Zha, X. Li, B. Yan, X. Jing, J. Li, and J. Xu. Incremental prediction model of disk failures based on the density metric of edge samples. *IEEE Access*, 7:114285–114296, 2019.

[13] Yu Wang, Qiang Miao, and M. Pecht. Health monitoring of hard disk drive based on mahalanobis distance. In *2011 Prognostics and System Health Managment Confernece*, pages 1–8, 2011.

[14] Y. Wang, Q. Miao, E. W. M. Ma, K. Tsui, and M. G. Pecht. Online anomaly detection for hard disk drives based on mahalanobis distance. *IEEE Transactions on Reliability*, 62(1):136–145, 2013.

[15] Jing Shen, Jian Wan, Se-Jung Lim, and Lifeng Yu. Random-forest-based failure prediction for hard disk drives. *International Journal of Distributed Sensor Networks*, 14(11), 2018.

[16] S. Huang, S. Fu, Q. Zhang, and W. Shi. Characterizing disk failures with quantified disk degradation signatures: An early experience. In *2015 IEEE International Symposium on Workload Characterization*, pages 150–159, 2015.

[17] Y. Xie, D. Feng, F. Wang, X. Tang, J. Han, and X. Zhang. Dfpe: Explaining predictive models for disk failure prediction. In *2019 35th Symposium on Mass Storage Systems and Technologies (MSST)*, pages 193–204, 2019.

[18] Y. Yi, J. Xiao, S. Wu, H. Li, and H. Jin. Failure order: A missing piece in disk failure processing of data centers. In *2019 IEEE 21st International Conference on High Performance Computing and Communications*, pages 223–230, 2019.

[19] Yong Xu, Kaixin Sui, Randolph Yao, Hongyu Zhang, Qingwei Lin, Yingnong Dang, Peng Li, Keceng Jiang, Wenchi Zhang, Jian-Guang Lou, Murali Chintalapati, and Dongmei Zhang. Improving service availability of cloud systems by predicting disk error. In *2018 USENIX Annual Technical Conference (USENIX ATC 18)*, pages 481–494, 2018.

[20] P. Anantharaman, M. Qiao, and D. Jadav. Large scale predictive analytics for hard disk remaining useful life estimation. In *2018 IEEE International Congress on Big Data (BigData Congress)*, pages 251–254, 2018.

[21] Jiang Xiao, Zhuang Xiong, Song Wu, Yusheng Yi, Hai Jin, and Kan Hu. Disk failure prediction in data centers via online learning. In *Proceedings of the 47th International Conference on Parallel Processing*, ICPP 2018, pages 35:1–35:10. ACM, 2018.

[22] C. A. C. Rincón, J. Pâris, R. Vilalta, A. M. K. Cheng, and D. D. E. Long. Disk failure prediction in heterogeneous environments. In *2017 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, pages 1–7, 2017.

[23] Farzaneh Mahdisoltani, Ioan Stefanovici, and Bianca Schroeder. Proactive error prediction to improve storage system reliability. In *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, pages 391–402. USENIX Association, 2017.

[24] N. Aussel, S. Jaulin, G. Gandon, Y. Petetin, E. Fazli, and S. Chabridon. Predictive models of hard drive failures based on operational data. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 619–625, 2017.

[25] I. C. Chaves, M. R. P. d. Paula, L. G. M. Leite, L. P. Queiroz, J. P. P. Gomes, and J. C. Machado. Banhfap: A bayesian network based failure prediction approach for hard disk drives. In *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 427–432, 2016.

[26] S. Ganguly, A. Consul, A. Khan, B. Bussone, J. Richards, and A. Miguel. A practical approach to hard disk failure prediction in cloud platforms: Big data model for failure management in datacenters. In *IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService)*, pages 105–116, 2016.

[27] W. Yang, D. Hu, Y. Liu, S. Wang, and T. Jiang. Hard drive failure prediction using big data. In *2015 IEEE 34th Symposium on Reliable Distributed Systems Workshop (SRDSW)*, pages 13–18, 2015.

[28] J. Li, X. Ji, Y. Jia, B. Zhu, G. Wang, Z. Li, and X. Liu. Hard drive failure prediction using classification and regression trees. In *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pages 383–394, 2014.

[29] Teerat Pitakrat, André van Hoorn, and Lars Grunske. A comparison of machine learning algorithms for proactive hard disk drive failure detection. In *Proceedings of the 4th International ACM Sigsoft Symposium on Architecting Critical Systems*, ISARCS '13, pages 1–10. ACM, 2013.

[30] Xiaoyi Sun, Krishnendu Chakrabarty, Ruirui Huang, Yiquan Chen, Bing Zhao, Hai Cao, Yinhe Han, Xiaoyao Liang, and Li Jiang. System-level hardware failure prediction using deep learning. In *Proceedings of the 56th Annual Design Automation Conference 2019*, DAC '19, pages 20:1–20:6, 2019.

[31] S. Basak, S. Sengupta, and A. Dubey. Mechanisms for integrated feature normalization and remaining useful life estimation using lstms applied to hard-disks. In *2019 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 208–216, 2019.

[32] F. D. d. S. Lima, G. M. R. Amaral, L. G. d. M. Leite, J. P. P. Gomes, and J. d. C. Machado. Predicting failures in hard drives with lstm networks. In *2017 Brazilian Conference on Intelligent Systems (BRACIS)*, pages 222–227, 2017.

[33] S. Pang, Y. Jia, R. Stones, G. Wang, and X. Liu. A combined bayesian network method for predicting drive failure times from smart attributes. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 4850–4856, 2016.

[34] C. Xu, G. Wang, X. Liu, D. Guo, and T. Liu. Health status assessment and failure prediction for hard drives with recurrent neural networks. *IEEE Transactions on Computers*, 65(11):3502–3508, 2016.

[35] F. Pereira, D. Teixeira, J. P. Gomes, and J. Machado. Evaluating one-class classifiers for fault detection in hard disk drives. In *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 586–591, 2019.

[36] L. P. Queiroz, F. C. M. Rodrigues, J. P. P. Gomes, F. T. Brito, I. C. Chaves, M. R. P. Paula, M. R. Salvador, and J. C. Machado. A fault detection method for hard disk drives based on mixture of gaussians and nonparametric statistics. *IEEE Transactions on Industrial Informatics*, 13(2):542–550, 2017.

[37] Lucas P. Queiroz, João Paulo Pordeus Gomes, Francisco Caio M. Rodrigues, Felipe T. Brito, Iago C. Chaves, Lucas Goncalves de Moura Leite, and Javam C. Machado. Fault detection in hard disk drives based on a semi parametric model and statistical estimators. *New Generation Computing*, 36:5–19, 2017.

[38] Y. Wang, E. W. M. Ma, T. W. S. Chow, and K. Tsui. A two-step parametric method for failure prediction in hard disk drives. *IEEE Transactions on Industrial Informatics*, 10(1):419–430, 2014.

[39] Y. Wang, K. Tsui, E. W. M. Ma, and M. Pecht. A fusion approach for anomaly detection in hard disk drives. In *Proceedings of the IEEE 2012 Prognostics and System Health Management Conference (PHM-2012 Beijing)*, pages 1–5, 2012.

[40] Xu Zhang, Qingwei Lin, Yong Xu, Si Qin, Hongyu Zhang, Bo Qiao, Yingnong Dang, Xinsheng Yang, Qian Cheng, Murali Chintalapati, Youjiang Wu, Ken Hsieh, Kaixin Sui, Xin Meng, Yaohai Xu, Wenchi Zhang, Furao

Shen, and Dongmei Zhang. Cross-dataset time series anomaly detection for cloud systems. In *2019 USENIX Annual Technical Conference (USENIX ATC 19)*, pages 1063–1076, Renton, WA, July 2019. USENIX Association.

[41] Y. Xie, D. Feng, F. Wang, X. Zhang, J. Han, and X. Tang. Ome: An optimized modeling engine for disk failure prediction in heterogeneous datacenter. In *2018 IEEE 36th International Conference on Computer Design (ICCD)*, pages 561–564, 2018.

[42] F. L. F. Pereira, F. D. d. S. Lima, L. G. d. M. Leite, J. P. P. Gomes, and J. d. C. Machado. Transfer learning for bayesian networks with application on hard disk drives failure prediction. In *2017 Brazilian Conference on Intelligent Systems (BRACIS)*, pages 228–233, 2017.

[43] Mirela Madalina Botezatu, Ioana Giurgiu, Jasmina Bogojeska, and Dorothea Wiesmann. Predicting disk replacement towards reliable data centers. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 39–48. ACM, 2016.

[44] J. Zhang, K. Zhou, P. Huang, X. He, M. Xie, B. Cheng, Y. Ji, and Y. Wang. Minority disk failure prediction based on transfer learning in large data centers of heterogeneous disk systems. *IEEE Transactions on Parallel and Distributed Systems*, 31(9):2155–2169, 2020.

[45] Charles X. Ling and Chenghui Li. Data mining for direct marketing: Problems and solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, KDD'98, page 73–79. AAAI Press, 1998.

[46] R. Larson. *Elementary Linear Algebra*. Cengage Learning, 2012.

[47] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

[48] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[49] Roger Lewis. An introduction to classification and regression tree (cart) analysis. 01 2000.

[50] Vladimir Svetnik, Andy Liaw, Christopher Tong, John Culberson, Robert Sheridan, and Bradley Feuston. Random forest: A classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43:1947–58, 11 2003.

[51] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[52] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[53] Joel Coffman and Alfred C. Weaver. Learning to rank results in relational keyword search. In *CIKM '11*, 2011.

[54] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[55] S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[56] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[57] Ying Zhao, Xiang Liu, Siqing Gan, and Weimin Zheng. Predicting disk failures with HMM- and hsmm-based approaches. In *Advances in Data Mining. Applications and Theoretical Aspects, 10th Industrial Conference, ICDM 2010, Berlin, Germany, July 12-14, 2010. Proceedings*, pages 390–404, 2010.

[58] H. Wang, A. Kläser, C. Schmid, and C. Liu. Action recognition by dense trajectories. In *CVPR 2011*, pages 3169–3176, 2011.

[59] Peter Prettenhofer and Benno Stein. Cross-language text classification using structural correspondence learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1118–1127, Uppsala, Sweden, 2010.

[60] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, CVPR '05, page 539–546. IEEE Computer Society, 2005.

[61] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging, 2015.

[62] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[63] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958, January 2014.

[64] Albert Orriols-Puig and Ester Bernadó-Mansilla. Evolutionary rule-based systems for imbalanced data sets. *Soft Comput.*, 13(3):213–225, 2009.

[65] D. Sun, Z. Wu, Y. Wang, Q. Lv, and B. Hu. Risk prediction for imbalanced data in cyber security : A siamese network-based deep learning classification framework. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, July 2019.

[66] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 103–119, Cham, 2018. Springer International Publishing.

[67] B. Zhu, G. Wang, X. Liu, D. Hu, S. Lin, and J. Ma. Proactive drive failure prediction for large scale storage systems. In *IEEE 29th Symposium on Mass Storage Systems and Technologies (MSST)*, pages 1–5, 2013.

[68] Ji Zhang, Yu Liu, Ke Zhou, Guoliang Li, Zhili Xiao, Bin Cheng, Jiashu Xing, Yangtao Wang, Tianheng Cheng, Li Liu, Minwei Ran, and Zekang Li. An end-to-end automatic cloud database tuning system using deep reinforcement learning. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, page 415–432, New York, NY, USA, 2019. Association for Computing Machinery.

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

[70] Ardeshir Raihanian Mashhadi, Willie Cade, and Sara Behdad. Moving towards real-time data-driven quality monitoring: A case study of hard disk drives. *Procedia Manufacturing*, 26:1107 – 1115, 2018. 46th SME North American Manufacturing Research Conference, NAMRC 46, Texas, USA.

[71] B. Eckart, X. Chen, X. He, and S. L. Scott. Failure prediction models for proactive fault tolerance within storage systems. In *2008 IEEE International Symposium on Modeling, Analysis and Simulation of Computers and Telecommunication Systems*, pages 1–8, 2008.

[72] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[73] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *CoRR*, 2017.