Future Generation Computer Systems 148 (2023) 460–471

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

SPAE: Lifelong disk failure prediction via end-to-end GAN-based anomaly detection with ensemble update

Yu Liu [c,a,1], Yunchuan Guan [a,1], Tianming Jiang [b,a,*], Ke Zhou [a], Hua Wang [a], Guangxing Hu [a], Ji Zhang [d], Wei Fang [d], Zhuo Cheng [d], Ping Huang [e]

[a] Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan, 430074, China
[b] School of Information Management, Central China Normal University, Wuhan, 430079, China
[c] School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan, 430074, China
[d] Innovation Center of Data Storage, HUAWEI Technology, 611730, Chengdu, China
[e] Temple University, Philadelphia, United States of America

ARTICLE INFO

Article history:
Received 19 September 2022
Received in revised form 27 March 2023
Accepted 20 May 2023
Available online 17 June 2023

Keywords:
Disk failure
Data reliability
SMART
Adversarial training
Ensemble update
Anomaly detection

ABSTRACT

Disk failure prediction aims to predict upcoming disk failures in advance for high data reliability. There are numerous supervised machine learning methods that are successful in predicting disk failure using SMART properties as input. However, these approaches heavily rely on a substantial number of annotated failed disks, resulting in degraded prediction performance caused by scarce failed disks at the beginning, also known as the cold start problem. Inspired by the success achieved in Generative Adversarial Network (GAN) based anomaly detection, this paper translates disk failure prediction into an anomaly detection problem. Specifically, we developed a **S**emi-supervised method for lifelong disk failure **P**rediction via **A**dversarial training and **E**nsemble update, called SPAE. The advantage of SPAE over existing supervised approaches is that SPAE can train the prediction model using only healthy disks, avoiding the cold start problem. Furthermore, SPAE can be updated using ensemble learning on emerging failed disks to resist the model aging problem. Compared to state-of-the-art methods using supervised machine learning on real-world datasets, SPAE predicts disk failures with higher accuracy for the full lifetime of models, *i.e.*, both the startup period and the long-term usage.

1. Introduction

To ensure high data reliability and availability of storage systems, some data redundancy schemes, *e.g.*, replication [1] and erasure coding [2,3], have been proposed and deployed in storage systems as remedy methods in response to disk failure occurrences. However, these approaches are reactive fault-tolerant techniques used to reconstruct data when disk failures occur, and thus they are storage space inefficient and bandwidth demands [4]. As a result, disk failure prediction has been proposed to predict disk failures before they occur. The key idea behind disk failure prediction is that if disk failures are predicted, users can be informed to take precautions, *e.g.*, data migration, which can significantly reduce maintenance costs.

Nowadays, the vast majority of modern hard disk drives have been equipped with Self-Monitoring, Analysis and Reporting Technology (SMART), which monitors individual disks and outputs attributes that contain information regarding the evolution of disk states. Therefore, SMART itself provides the ability to predict impending disk failures. Specifically, before a disk fails, a binary warning (will fail) will be issued if any attribute exceeds its threshold defined by the manufacturers [5]. However, this threshold-based method can only reach a failure detection rate (FDR) of 3%–10% with 0.1% false alarm rate (FAR) [6] due to its simplicity and the conservative settings of thresholds [7]. Note that FDR equals the recall rate in this paper.

Supervised machine learning methods have been introduced to improve the FDR [6–15]. These approaches take the SMART attributes as input, followed by classifiers implemented using supervised machine learning algorithms. Although some approaches [7,14,15] have achieved high FDRs and low FARs, they suffer from the data imbalance issue, *i.e.*, the number of failed disks is much smaller than that of healthy ones. In addition, the training data is gradually gathered instead of being given in advance [7]. As a result, the training data, especially for failed disks, collected within the initial period may be insufficient, resulting in an inability of the predictor at the beginning of its deployment, *i.e.*, the cold start problem.

Observing the emergence pattern of failed disks, we find the same problem with anomaly detection in the field of computer

* Corresponding author at: School of Information Management, Central China Normal University, Wuhan, 430079, China.
E-mail address: tmjiang@ccnu.edu.cn (T. Jiang).

[1] Yu Liu and Yunchuan Guan contribute equally to this work and should be considered the co-first authors.

https://doi.org/10.1016/j.future.2023.05.020
0167-739X/

vision, *i.e.*, there is a wide gap in the ratio between normal and abnormal samples. Recent research has demonstrated that Generative Adversarial Networks (GANs) are able to capture the data distribution and thus have been investigated for anomaly detection. Referring to the success story of anomaly detection in dealing with the heavily unbalanced dataset and inspired by the significant success achieved by GAN-based approaches in anomaly detection, in this paper, we propose a novel end-to-end deep learning method for detecting impending disk failures. To cope with the early absence of failed disks, our approach can complete the learning of the detection model using only healthy disks. Furthermore, as failed disks continue to emerge, our model can be updated with ensemble learning to alleviate the problem of model aging [7] and maintain stability in prediction accuracy. Therefore, our **S**emi-supervised method for lifelong disk failure **P**rediction via **A**dversarial training and **E**nsemble update, SPAE, is applied to practical failure prediction by disk SMART attributes.

However, it is tricky to apply GAN-based anomaly detection for disk failure prediction. The first challenge is how to transfer non-image SMART data into 2D image-like representation data for the deployment of deep learning techniques. We tackle this challenge with a novel data construction method that segments time-series SMART data via a sliding window. The merit of the image-like representation data lies in that it enables the deployment of deep learning techniques and automatic feature extraction, killing two birds with one stone.

Our contributions are described below.

- To the best of our knowledge, we pioneer the use of GAN-based anomaly detection for disk failure prediction to deal with the cold start problem.
- Based on the GAN's construction and loss functions, we pioneer the use of ensemble learning for prediction model updates to deal with the aging problem.
- We propose a novel data construction pattern, transforming the SMART data into 2D image-like representations and enabling the deployment of deep learning techniques with automatic feature extraction.
- Conducting experiments on real-world datasets, we simulate the use of our model in both the initial period and the long-term use and validate its applicability for large- and small-sized datasets. The experimental results demonstrate the effectiveness of our model.

The rest of this paper is organized as follows: In Section 2, we introduce the motivations of our work. Section 3 describes the proposed approach. Our experimental results are discussed in Section 4. The related work is shown in Section 5. The last section is conclusions and future work.

2. Background and motivations

2.1. Background

Since hard disk drive failures account for 78% of the hardware replacements in data centers [16], they have become a common occurrence as storage sizes continue to increase [1,17]. In addition, if not handled properly, they can result in catastrophic consequences, causing not only service downtime but also data loss if no data redundancy schemes are in place [18,19]. By predicting possible disk failures and enabling early updates of failed disks, the stability of the data center can be effectively ensured. There are two main challenges that need to be adequately addressed for high prediction performance, *i.e.*, the data imbalance issue and the time series feature extraction from SMART data.

2.2. Data imbalance issue

Since disk failure is relatively a rare event and failed disks only account for a tiny part of all disks [7], there exists a data imbalance phenomenon, *i.e.*, the number of healthy disks is much larger than that of failed ones. However, traditional supervised machine learning methods work well only for balanced datasets [20]. When working on an imbalanced dataset, these methods biasedly predict all disks to be healthy for overall high accuracy [20].

To this end, some re-balancing techniques [7,13], *i.e.*, oversampling and downsampling, are proposed to obtain a balanced training set for these supervised methods. Botezatu et al. [13] downsample the healthy samples of the entire training set to an amount that is close to the size of the failed samples. In [7], the online bagging technique is used to sample the sequentially arrived data for online learning. Besides the downsampling scheme, there are also works that address the data imbalance issue from the cost-sensitive learning perspective [15], assigning different costs to the false positive (FP) and false negative (FN).

2.3. Time series features

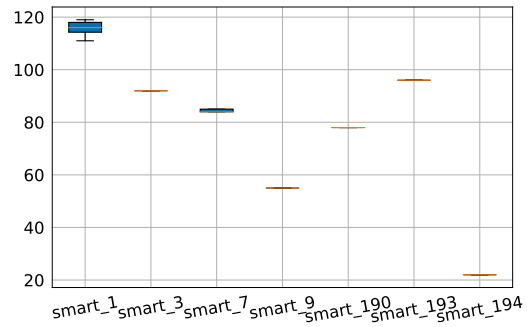
1D-SMART attributes (the SMART attributes of a disk at a specific timepoint) are fed into classifiers implemented by supervised machine learning algorithms while the changes of SMART attributes over time are ignored. Previous research has shown that the changes in SMART attributes over time, *i.e.*, temporal locality, are beneficial to distinguishing disk failures. In [14], Zhu et al. calculate the absolute differences between the current SMART attributes values and their corresponding values six hours ago as time series features. Formally, given a time window size of w, the absolute difference at timestamp t is noted as *Diff* and calculated as $Diff(x, t, w) = x(t) - x(t - w)$ [14,15]. Besides *Diff*, *Var* means the variance of attribute values within a period and is calculated as $Var(x, t, w) = E[(X - \mu)^2]$, and *Bin* means the sum of attribute values within a period and is calculated as $Bin(x, t, w) = \sum_{j=t-w}^{t} x(j)$ [15].

To validate the importance of time series features in SMART data, we conduct two preliminary examinations. We calculate the variance of each SMART attribute within a period and sum these variances up. The results reveal that failed disks score higher than healthy disks in terms of the sum of variances. As shown in Fig. 1, we depict a part of SMART attributes within 30 days for a healthy disk and a failed disk by the box plots. It is clear that compared to the attributes of the healthy disk, the counterparts of the failed disk fluctuate drastically. In summary, time series features can indicate disk failure and play an essential role in predicting such failures.

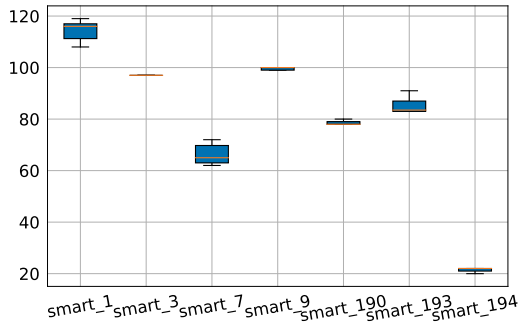
Although RNN and LSTM are commonly used to predict time series, the training and predicting process of these models is hard to be parallelized and thus suffers from high overhead [21], especially in the case of long time series. To enable the deployment of deep learning techniques with low overhead, 2D-SMART attributes are taken as input. 2D-SMART attributes refer to SMART attributes of a disk within a period and are obtained by stacking several 1D-SMART attributes within a specified period. This construction renders image-like representations that are suitable for the convolution operations of deep learning. Besides enabling the deployment of deep learning, therefore, 2D-SMART attributes also bring the benefit of automatic feature extraction. We illustrate the construction of 2D-SMART attributes in detail in Section 3.1.

3. The proposed method

Our goal is to predict whether a disk will fail within a given time interval using the SMART data reported by the disk. For



(a) healthy disk



(b) failed disk

Fig. 1. The box plots of the SMART attributes of a healthy disk and a failed disk within 30 days. The *y*-axis is the normalized SMART value. The orange horizontal line of each box represents its median value. Longer box lengths indicate larger data fluctuations.

the sake of simplicity in the following discussion, we constrain the period to seven days before a faulty event. We formulate the prediction problem as an anomaly detection problem instead of a classical binary classification problem.

In this section, we will describe the framework of SPAE and its training techniques. As shown in Fig. 2, SPAE is a failure prediction model with an ensemble approach for model updating. As its backbone, SPA [22] comprises two main components: (1) the data processing and (2) the training of GAN-based disk failure prediction.

3.1. Data processing

3.1.1. Normalization

Since different SMART features will fall into different value domains, *e.g.*, SMART1 is often greater than 1 000 000 while the maximum value of SMART9 is 100, we apply data normalization to ensure a fair comparison. In general, SMART attributes of failed disks fluctuate drastically. However, some SMART attributes are constant and only jump once before disk failure. It is hard to distinguish long-failed disks and healthy disks by these SMART attributes using local normalization. As a result, we use global min–max normalization [7,15] to regular attributes. The normalization is shown below.

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{1}$$

where x is the original value of a feature, and *max* and *min* are the maximum value and the minimum value of the feature in our dataset, respectively.

3.1.2. Construction of 2D-SMART attributes

To enable a GAN-based model for disk failure prediction, we must reformat the SMART attributes as input. Inspired by [23,24] where a 2D chunk, *i.e.*, image-like representation, is adopted as the input of Convolutional Neural Network (CNN), we reformat the 1D-SMART attributes into 2D input chunks which maintain the temporal locality of the time series SMART data. As shown in Fig. 3, 1D-SMART attributes refer to the M selected SMART features of a disk at a specific timepoint, and 2D-SMART attributes indicate a group of fixed order 1D-SMART attributes within a T time range. The construction denoted as 1D-to-2D in the following discussion, of 2D-SMART attributes exploits CNN's advantage of feature extraction, which is automatically done and adapted by the deep learning model itself. Note that since the 1D-SMART attributes have no sequential relationship, our model uses a 1D convolution kernel to convolve temporal features when convolving the 2D-SMART attributes. In addition to our naive transformation, MTF(Markov Transition Fields) and GTF(Gramian Transition Fields) [25] are also worthy of being noticed. However, since these methods transform each 1D-SMART time series into a 2D matrix, the transformation of multiple SMART time series will yield 3-dimensional data. As a result, the overhead costs incurred by these methods may outweigh the gains in accuracy. The comparison will be detailed in Section 4.4.1.

3.2. GAN-based method for disk failure prediction

3.2.1. GAN for anomaly detection

The framework of Generative Adversarial Networks (GANs) [26] consists of two components: the generator G, which aims to produce synthetic data that resembles real training data, and the discriminator D, which aims to distinguish between real training samples and fake data produced by G. The key idea behind GAN is that G and D compete with each other like two players in a game, and are optimized alternatively using stochastic gradient descent (SGD). Formally, the competition between the generator G and the discriminator D can be computed as follows:

$$\min_G \max_D \mathbf{E}_{x\sim p_d(x)}\left[\log D(x)\right] + \mathbf{E}_{x'\sim p(x')}\left[\log(1 - D(G(x')))\right], \tag{2}$$

where x is a training sample abiding by the true data distribution $p_d(x)$, and x' is a latent vector sampled from a prior distribution $p(x')$ (*e.g.*, standard normal distribution).

Recently, Akcay et al. [27] established a generic GAN-based anomaly detection framework that is made up of encoder–

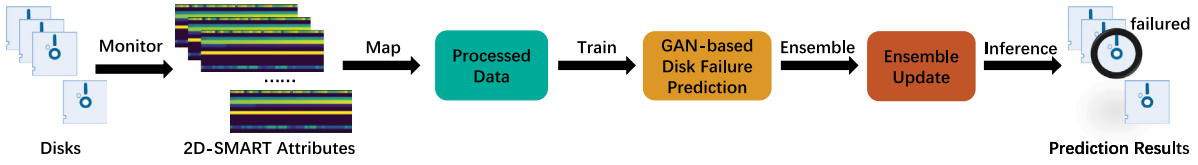


Fig. 2. Overview of SPAE framework and its three main components, *i.e.*, the data processing, the GAN-based method for disk failure prediction, and ensemble update. After the SMART attributes are monitored, they are aggregated and mapped to processed data, which are image-like representations. Then, in the training phase, the healthy image-like representations are used to train a GAN-based method for disk failure prediction via adversarial training. To deal with the model aging problem, SPAE uses the emerging SMART data for ensemble updates.

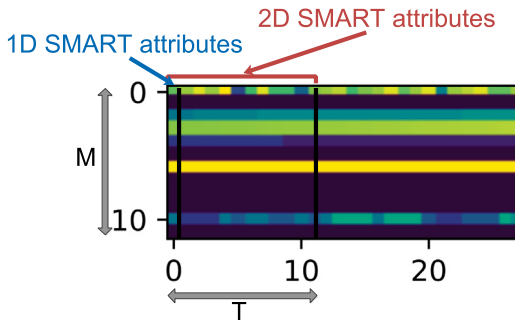


Fig. 3. Construction of 2D-SMART attributes. For each individual disk, we first stack its continual 1D-SMART attributes (the SMART attributes of a disk at a specific timepoint) and then segment data using a sliding window with the same size of selected features, resulting in 2D-SMART attributes (M selected features within time range T, *i.e.*, the size is $M * T$).

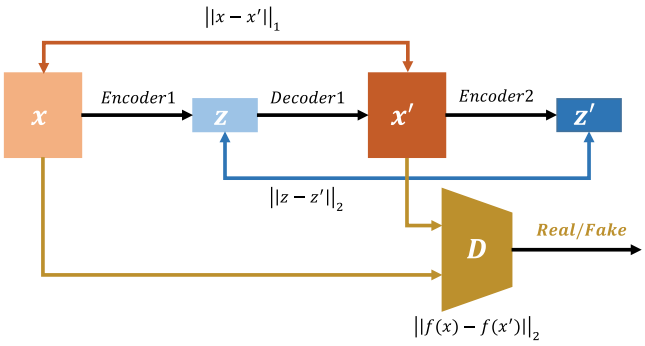


Fig. 4. *Encoder*1 learns the input data representation z and *Decoder*1 reconstructs the input. *Encoder*2 learns the representation z' of reconstructed data. In the training phase, the model learns both the normal data distribution and minimizes the output anomaly score L_1, calculated as $\|z-z'\|_1$. In the test phase, the anomaly score L_1 is compared with a threshold ϕ, and an anomaly will be alarmed if $L_1 > \phi$.

decoder–encoder sub-networks and achieves satisfactory performance for anomaly detection problems. They take normal samples as input and use an autoencoder in standard GAN to generate samples that are as close as possible to normal samples. The details of their GAN-based anomaly detection are shown in Fig. 4. They define the normal sample x as real data and the reconstructed sample x' as fake data. The autoencoder network for image generation can learn the feature representation z of the input sample x. To detect anomalies, they add an encoder to learn the representation z' of the reconstructed sample x'.

During the training phase, G and D are optimized alternately. The optimized loss of D is defined as

$$L_D = L_B(D(x), y) + L_B(D(x'), y'), \tag{3}$$

where $L_B(\cdot)$ denotes the Binary Cross-Entropy function, y and y' represent the labels of x and x', respectively. The optimized loss of G is defined as

$$L_G = \alpha L_1 + \beta L_2 + \gamma L_3, \tag{4}$$

where $L_1 = \|z - z'\|_2$, $L_2 = \|x - x'\|_1$, and $L_3 = L_B(D(x'), y)$, while α, β, and γ are hyperparameters. Following [27], we set $\alpha = 1$, $\beta = 50$, and $\gamma = 1$.

During the anomaly detection phase, the difference between z and z' is used to measure the effectiveness of sample generation. The smaller the difference is, the better the sample generation is. The model learns the distribution of normal samples during training since only normal samples are used. As a result, the difference between normal samples becomes smaller. On the other hand, the abnormal sample deviates from the normal sample distribution, making the difference more significant. Therefore, the difference L_1 can be used to detect anomalies. If its value exceeds a certain threshold, the sample is considered abnormal.

Algorithm 1 Model-updating-enabled GAN-based Algorithm

Input: Disk identifier: i; Current 1D-SMART attributes: $\vec{x}$; Current disk status: y
Input: 1D-SMART attributes Dataset: S; Constructed 2D-SMART attributes dataset: $S^{'}$
Output: Prediction result: $y^{'}$

```
//Model update phase
if y == 1 then                                   ▷ Disk D_i is failed
    deleteDisk(D_i)
else                                             ▷ Disk D_i is operating
    if ifFull(Q_i) then
        x⃗' ← dequeue(Q_i)
        enset(S, x⃗')
    end if
    enqueue(Q_i, x⃗)
    if ifFull(S) then
        S' ← 1D − to − 2D(S)                     ▷ construct 2D-SMART attributes samples using dataset S
        fine-tune(oldGAN, S')                    ▷ fine-tune old GAN using dataset S'
        emptyset(S)                              ▷ empty dataset S
    end if
    //Prediction phase
    X ← 1D − to − 2D(Q_i)                        ▷ construct 2D-SMART attributes samples using Q_i
    y' ← predictGAN(X)
    if y' == 1 then                              ▷ Disk D_i is soon-to-fail
        Trigger an alarm
    end if
end if
```

3.2.2. GAN for disk failure prediction

In our disk failure prediction scenario, the underlying distribution of SMART attributes gradually changes with time [7,11]. As a result, we encounter the model aging problem, *i.e.*, the prior trained model will lose validity on the new SMART data. To deal with the model aging issue, we use the fine-tuning feature of CNN to do model updating. Fine-tuning is a common technique to transfer information from one dataset to another. Unlike the 1-month replacing strategy [11], which discards the old model and trains a brand-new model using new data, fine-tuning belongs to the accumulation updating strategy [11], which reuses the old model and retrains it on new data.

However, sample labeling is very challenging due to the training samples arriving continuously and the statuses of disks being uncertain [7]. To address this issue, we adopt the automatic online labeling method proposed by Xiao et al. [7]. In detail, a fixed length first-in–first-out queue Q_i is used to store the samples for disk D_i and keep the samples unlabeled. After D_i has failed, all the samples in the queue Q_i will be labeled as positive. If D_i is still in operation, Q_i outputs the oldest samples that are then labeled as negative and replaced with new ones.

Unlike the supervised method used in [7] where both healthy and failed samples are used to train the models, our semi-supervised method only uses healthy samples. Another difference is that fine-tuning reduces the updating frequency. We update our model using batches of samples instead of every single new sample used in online learning [7]. Specifically, we use a dataset

Table 1
The 24 used SMART attributes.

Attribute ID	Attribute name
1	Raw Read Error Rate
3	Spin Up Time
4	Start/Stop Count
5	Reallocated Sectors Count
7	Seek Error Rate
9	Power-On Time Count
10	Spin up Retry Count
12	Power Cycle Count
183	SATA Downshift Error Count
184	I/O Error Detection and Correction
187	Reported Uncorrectable Errors
188	Command Timeout
189	High Fly Writes
190	Airflow Temperature
191	G-sense error rate
192	Power-Off Retract Count
193	Load/Unload Cycle Count
194	Temperature
197	Current Pending Sector Count
198	Offline Uncorrectable Sector Count
199	Ultra ATA CRC Error Rate
240	Head Flying Hours
241	Total LBAs Written
242	Total LBAs Read

S to maintain annotated data within a constant time interval and update models using dataset S. When dataset S is full, we construct them into 2D-SMART attribute chunks, *i.e.*, image-like representations, as shown in Fig. 3. Note that in our implementation, the model updating interval is not equal to the prediction time interval, and we predict for each sample currently collected. The proposed model-updating-enabled GAN-based algorithm for disk failure prediction is illustrated in Algorithm 1. The time complexity of Algorithm 1 is $O(NMT)$, where N denotes the number of samples in dataset S, M denotes the number of SMART attributes in 2D-SMART attributes and T denotes the time range of 2D-SMART attributes.

3.3. SPAE: Extension of SPA

SPAE extends SPA on three aspects: (1) On the input side, SPAE uses all SMART attributes to achieve end-to-end learning without feature selection. (2) On the output side, SPAE adopts an ensemble approach to achieve higher and more stable prediction performance. (3) SPAE enables leveraging of all data ever collected.

3.3.1. End-to-end learning mechanism

Feature selection as basic processing for disk failure prediction ensures the low overhead and high accuracy of traditional machine learning models [7] (*e.g.*, support vector machines (SVM)). Nevertheless, this processing seems to be redundant in deep learning because the prerogative of deep learning is to automatically pick the most representative features [28]. In fact, inputting all features into a deep model for end-to-end learning has the opportunity to yield better results, as the model may find available information that humans consider irrelevant for prediction. Although higher dimensions of features mean more training time, the extra training time is less than the time of feature selection. Therefore, we improve our method by the end-to-end learning mechanism instead of feature selection. In this paper, we use all SMART attributes provided by Backblaze [29]. For each attribute, it contains two values of interest, including a raw value and a normalized value. As a result, we have 48 features as the input for the classifier. The used SMART attributes are shown in Table 1.

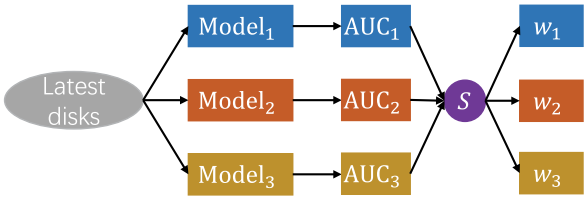


Fig. 5. The process of ensemble update. W_1, W_2, and W_3 are the coefficients/weights of sub-models generated by the learning on the failed disks. The updated model will determine the state of disks based on voting results from all sub-models.

3.3.2. Ensemble update

As mentioned in 3.2, SPA focuses on the sensitivity of the reconstructed vector z to healthy samples, where only a sub-model consisting of *Decoder*1 and *Encoder*2 is involved in anomaly detection. In fact, after training GAN, the reconstruction ability of the sub-model consisting of *Encoder*1 and *Decoder*1 as well as the discriminator ability of the sub-model consisting of D are also improved. As a result, the two differences, *i.e.*, L_2 and $L_3' = \|f(X) - f(X')\|_2$, generated by the above two sub-models are both sensitive to healthy samples and can be used for anomaly detection, where $f(\cdot)$ is a function used for feature extraction. $D(X) = F(f(X))$, where $F(\cdot)$ is a function that changes the vector, *i.e.*, feature, to a scalar. Since ensemble learning methods, *e.g.*, Random Forest [30] and Isolation Forest [31], can aggregate multiple classification models to achieve higher and more stable prediction performance than a single model, we use an ensemble method to combine the three sub-models of SPA to enhance anomaly detection.

Specifically, as shown in Fig. 5, we evaluate the anomaly detection capability of each sub-model, using the latest disks as the validation set. We use *area under curve* (AUC) [32] as the evaluation metric because it considers the model's prediction performance for both failed disks and healthy disks. We use the Softmax function to transform the AUC value of each sub-model into its ensemble weight. Based on these weights, *i.e.*, w_1, w_2, and w_3, we define the final loss below.

$$L = w_1\|z - z'\|_2 + w_2\|x - x'\|_1 + w_3\|f(x) - f(x')\|_2, \tag{5}$$

where a failed disk will be alarmed if $L > \phi$. The determination of threshold ϕ will be described in Section 4.2.

3.3.3. Update with both healthy and failed disks

Although SPAE was designed to not rely on failed disks for detection, it is sensible to use them as they emerge. Thus, we have updated the pattern to use both healthy and failed disks in SPAE. In the implementation, we first use the healthy disks to train and fine-tune the model to maintain the feature awareness of the initial healthy disk. Then, we gather the failed disks and use both healthy and failed disks to update the model using an ensemble learning model. Note that we can also update the model without failed disks, but it may result in a slight loss of prediction performance.

4. Experimental results

In this section, we design experiments to verify the superiority of SPAE and the benefits of end-to-end learning mechanisms and ensemble updates. Note that we refer to the SPAE technique without an end-to-end learning mechanism and ensemble update as SPA.

Table 2
Hyperparameters setup.

Method	Hyperparameters					
RF	n_tree = 150					
SVM	kernel = linear	C = 8	gamma = 0.1			
BP	hidden layer = 64	activation = ReLU	epoch = 1000	learning rate = 0.01	optimizer = Adam	
DeepSVDD	hidden layer = (256,128)	activation = ReLU	epoch = 100	learning rate = 0.001	batch size = 1024	out_dim = 128
IForest	n_tree = 120	max features = 1.0	max samples = 2000			
SPA	z_dim = 100	image size = 12				
SPAE	z_dim = 100	image size = 12				

Table 3
Overview of dataset.

Dataset	Disk model	Class	No. disks
STA (large-size)	ST4000DM000	Good	33,701
		Failed	1508
STB (small-size)	ST8000DM002	Good	9887
		Failed	92
STC (small-size)	ST8000NM0055	Good	14,421
		Failed	88

4.1. Experimental setup

4.1.1. Experimental platform

The experimental evaluation is performed on a Ubuntu 18.04 server with Intel Xeon Silver 4210R CPU, 128 GB Memory, and NVIDIA RTX 3090 graphics card. The proposed SPAE is implemented by PyTorch 1.10.0.

4.1.2. Datasets

To assess the proposed method, we utilized public datasets from Backblaze [29]. These datasets cover a 12-month period from January 2017 to December 2017 and include daily snapshots of all SMART attributes for each operational disk. We specifically selected three disk models from the datasets: Seagate's ST4000DM000, ST8000DM002, and ST8000NM0055. The first dataset contains more failed disks than the latter two. A disk is considered failed if it was replaced as part of a repair procedure, which we inherited from Pinheiro et al. [33]. Table 3 displays all disks that failed and were replaced in 2017.

To address the data imbalance issue in supervised methods, we use a common under-sampling technique [34] to reduce the number of majority class samples. This results in varying ratios of healthy to failed samples, ranging from 1:1 to 1:50. In our final training set, we set the ratio to 1:5, which leads to higher prediction accuracy. Note that we use the same training set to train both SPA and SPAE models, but only healthy samples from the dataset are used.

4.2. Metrics

We use the FDR and the FAR metrics, which have been widely used to evaluate the effectiveness of disk failure prediction [7,14]. FDR is defined as the ratio of correctly predicted failed disks to the total number of failed disks:

$$FDR = \frac{\#\text{true positives}}{\#\text{true positives} + \#\text{false negatives}} \tag{6}$$

FAR is defined as the ratio of mispredicted healthy disks to the total healthy disks:

$$FAR = \frac{\#\text{false positives}}{\#\text{false positives} + \#\text{true negatives}} \tag{7}$$

Note that a disk is predicted as failed if any of the samples from it is predicted as failed. Moreover, a failed disk is correctly detected only when any of the samples collected within the last week before failure is predicted positive, and a healthy disk is mispredicted if any of the samples collected outside the latest week is predicted as positive [7]. We calculated FDRs and FARs in each model updating interval. Referring to [7], we measured the FDRs under the constraint that the FARs are around 1.0% as well as the FARs under the constraint that the FDRs are around 85%.

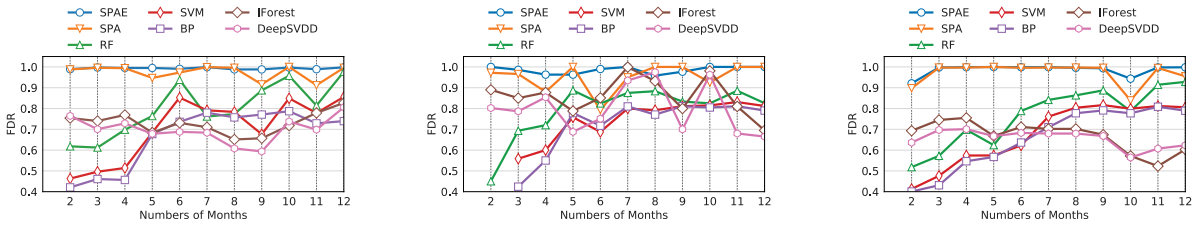
4.3. Comparisons with existing methods

For the purpose of simulating real application scenarios, we divide the disk samples into a training set and a test set in chronological order to ensure that the test set samples are strictly later than the training set samples. To demonstrate the effectiveness of our model, we compared SPAE with SPA, three commonly used supervised classification algorithms, and two unsupervised classification algorithms. The supervised classification algorithms include RF [35], SVM [36], and BP (backward propagation neural networks). The unsupervised algorithms consist of IForest [37] and DeepSVDD [38]. Note that RF is demonstrated to be able to deliver state-of-the-art performance for disk failure prediction [7,15]. As shown in Table 2, for RF, we experimented with different numbers of trees and settled on using 150 trees because of their superiority in performance. For SVM, we use the LIBSVM library [39], and select polynomial kernel. For BP, we use three layers of BP with 64 nodes in the hidden layer and use *ReLU* as the activation function. We set the maximum number of iterations to 1000, the learning rate to 0.01, and adopted Adam [40] for optimization. For IForest, we use the version from the PYOD library [41]. We set the number of trees as 120, max features as 1.0, and max samples as 2000. For DeepSVDD, we use the version from the DeepOD library [42]. We set the width of two hidden layers as 256 and 128, activation function as *ReLU*, epoch as 100, learning rate as 0.001, batch size as 1024, and output dim as 128. All the mentioned hyperparameters are selected by 3-fold cross validation [43] and the rest hyperparameters are set as default values. For SPA and SPAE, we build the scheme upon the code from [27] and set the size of z as 100. For the square image-like representation is the commonly used input shape for CNN, we set $T = M = 12$.

To address the issue of model aging, we update both the parameters of the neural network and the weights of loss functions. We fine-tune the parameters with a sampled training dataset of healthy samples collected in the previous month. For updating the weights, we calculate them using validation samples collected in the latest month. Then, we evaluate the model's prediction performance on the test set every month.

We set the model updating interval as one month to ensure a fair comparison with the accumulation update strategy used in [7,11]. In those strategies, all the data collected from the beginning is used to update the offline models once a month. For each month, we build offline models with all the training data collected so far and investigate their performance on the same test set.

As shown in Fig. 6, we show FDRs on STA, STB, and STC, respectively, where the FAR is less than 0.01. From the first three

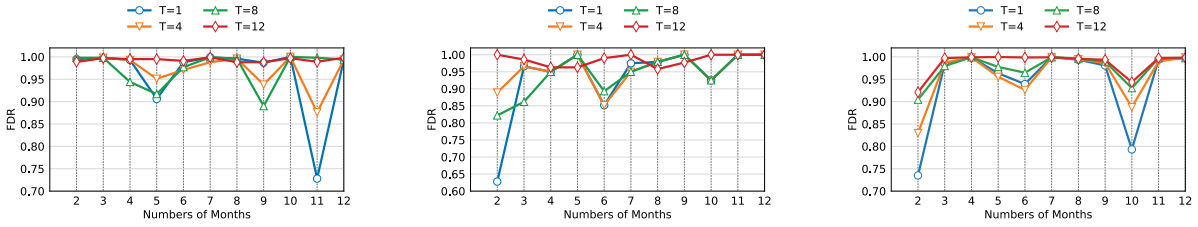


(a) STA (b) STB (c) STC

Fig. 6. Comparisons of FDRs with existing methods on STA, STB, and STC.

Table 4
Average FDRs and FARs of different disk failure prediction methods on STA, STB, and STC in 2017.

Method		SPAE			SPA			RF			BP			SVM			IForest			DeepSVDD		
Dataset		STA	STB	STC	STA	STB	STC	STA	STB	STC	STA	STB	STC	STA	STB	STC	STA	STB	STC	STA	STB	STC
Average	FDR	0.993	0.985	0.985	0.974	0.953	0.970	0.800	0.791	0.766	0.665	0.727	0.658	0.704	0.746	0.678	0.730	0.861	0.668	0.699	0.799	0.655
Average	FAR	0.001	0.001	0.002	0.003	0.004	0.004	0.013	0.017	0.024	0.033	0.029	0.035	0.036	0.029	0.032	0.029	0.011	0.031	0.030	0.019	0.046



(a) STA (b) STB (c) STC

Fig. 7. FDRs under different time range T on STA, STB and STC.

months, we observe a similar phenomenon shown in [44] that all the supervised methods exhibit poor performance due to the fact of lacking ample failed disks. Obviously, they suffer from the cold start problem. In contrast, the unsupervised methods (IForest, DeepSVDD, SPA, and SPAE) achieve higher FDRs from the beginning, which demonstrates our model is effective in alleviating the cold start problem. We attribute it to merely training the healthy samples. In addition, for the long-term effect, SPAE outperforms its supervised and unsupervised counterparts. We believe that, on the one hand, anomaly detection schemes are able to detect the unseen anomaly case [27] compared to supervised schemes. On the other hand, ensemble learning enhances prediction performance. The average FDRs and average FARs are detailed in Table 4. Note that there are differences in the performance of the prediction methods on the three datasets. We attribute it to the differences in the failure rates and the features of SMART data across different disk models.

4.4. Ablation study

4.4.1. The effectiveness of 2D image-like representation

To evaluate the effectiveness of the proposed 2D image-like representation, we train models with different time ranges, including 1, 4, 8, and 12 in units of days. Note that SMART attributes are collected on a daily basis. $T = 1$ indicates the specific case of 1D-SMART attributes, *i.e.*, no time series data is used. In addition, we chose a suitable time window size from 4, 8, or 12. Fig. 7 shows the FDRs on the datasets STA, STB, and STC, respectively. Obviously, the model trained with $T = 1$ achieves satisfactory performance, which demonstrates the effectiveness of our adversarial learning strategy. However, it is consistently outperformed by the models trained with other values of T in both figures. These results demonstrate the effectiveness of 2D image-like representations because they exploit the inherent time series features of SMART data. When comparing the performance under different values of T, we observe that the models trained with $T = 12$ consistently outperform the ones trained with other values of T. As a result, we set $T = 12$ in the following experiments.

Table 5
Comparison of the FDR and time overhead between MTF-SPAE and SPAE. Training time and testing time are calculated with 100 samples/ms.

Method	Dataset	FDR	Training time (ms)	Testing time (ms)
SPAE	STA	0.996	746.8	7.174
	STB	0.985		
	STC	0.991		
MTF-SPAE	STA	0.993	28 677.1	243.8
	STB	0.985		
	STC	0.986		

As mentioned in 3.1.2, although introducing MTF and transforming the SMART series into 2D-matrices seems to be a prospective method, it will bring significant cost since we use all SMART features. As shown in Table 5, we count the average FDR of MTF-SPAE (SPAE with MTF transforming) and ordinary SPAE (SPAE with SMART series stacking) from February to December on the three disk models. Meanwhile, we record the training and prediction time of each method. We find that compared to the cost in overhead, the improvement of introducing MTF in accuracy is small. We believe that introducing MTF would obtain significant benefits under an acceptable overhead if we can select a small number of SMART attributes. However, it conflicts with the demand for the end-to-end mechanism. As a result, we chose the simple but efficient transformation, *i.e.*, stacking 1D-SMART attributes.

4.4.2. The effectiveness of end-to-end mechanism

To evaluate the effectiveness of the end-to-end mechanism, we compared SPAE using all features with the method using the picked features by selection processing in terms of FDR. We first eliminate all SMART attributes with a null value for methods.

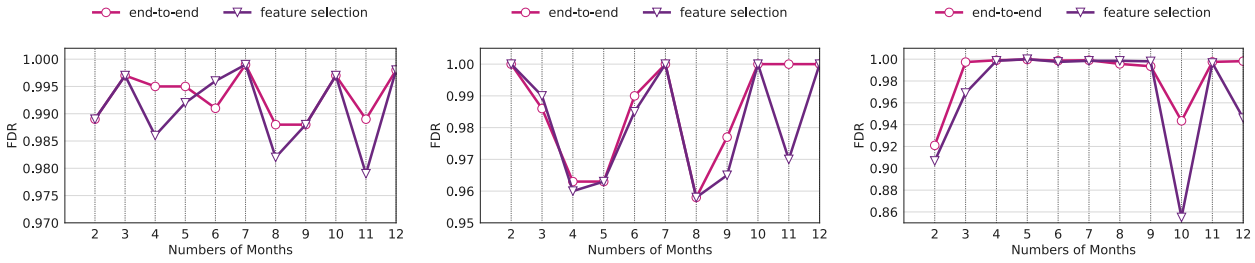


Fig. 8. FDRs of SPAE using the end-to-end mechanism or the feature selection on STA, STB, and STC.

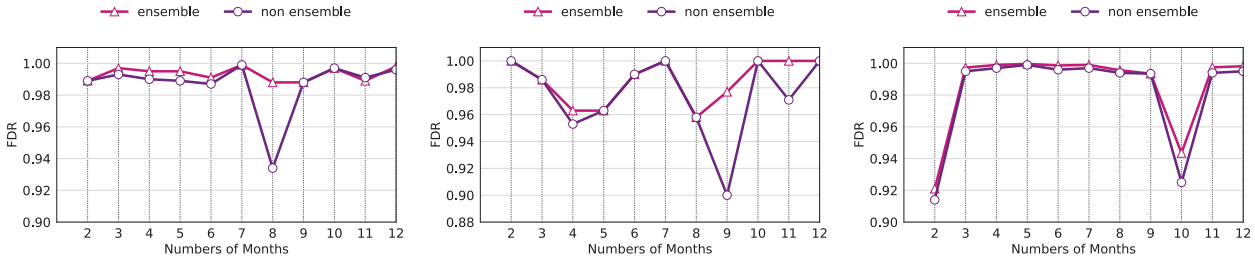


Fig. 9. FDRs under the ensemble update or not on STA, STB, and STC.

Table 6
Comparison of the time overhead between end-to-end mechanism and feature selection. Training time and testing time are calculated with 100 samples/ms.

Method	Preprocessing time	Training time (ms)	Testing time (ms)
End-to-end method	No feature selection	746.8	5.265
Method based on feature selection	Additional feature selection time	754.9	5.198

Fig. 8 shows the FDRs on STA, STB, and STC, respectively. It is clear that the end-to-end mechanism slightly outperforms the feature selection processing on STA, while it is neck-to-neck on STB and STC. We believe that, on the one hand, SPAE is a deep model, which can automatically perform feature selection during training. On the other hand, the end-to-end mechanism provides richer information from more SMART features. Note that we get opposite results for the data points of June in Fig. 8(a), March in Fig. 8(b), and September in Fig. 8(c). We attribute it to fluctuations in model prediction performance caused by different features of disks in different months. In addition, it can be seen that the feature selection scheme is only about 0.5% higher than the end-to-end scheme in these data points in terms of FDR.

In terms of time overhead, we compare the time overhead of the end-to-end mechanism and the method based on feature selection in the phases of preprocessing, training, and testing. As shown in Table 6, on the one hand, these methods have almost the same time overhead in the phase of training and testing. The reason is that the two methods differ only in the size of the input 2D image-like representation, where SPAE would normalize the size of input images before putting them into the neural network. On the other hand, in the preprocessing phase, the end-to-end mechanism bypasses the time-consuming and manually involved feature selection. In summary, the end-to-end mechanism is better than the hand-crafted feature selection for SPAE.

4.4.3. The effectiveness of ensemble method

To evaluate the effectiveness of the ensemble update in SPAE, we compared SPAE with SPA in FDR. As shown in Fig. 9, SPAE gets higher and more stable results than that of SPA on three datasets. We believe that the ensemble learning approach combines multiple models and reduces the variance of prediction performance [45]. In addition, SPA can be seen as a special case for SPAE (L_1's weight is set to 1 and the other weights are set to 0). With the help of the validation set, SPAE can learn weights considering the effect of both healthy and failed samples to achieve superior prediction performance.

Table 7
Comparison of the time overhead for the ensemble update and the non-ensemble update. Testing time is calculated with 100 samples/ms.

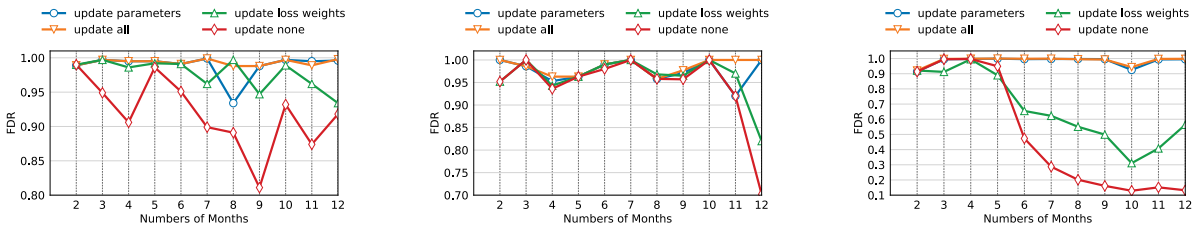
Method	Validating time (s)	Testing time (ms)
Ensemble	7.639	7.170
Non-ensemble	0	5.265

In terms of time overhead, ensemble update and non-ensemble update differ in validating phase and testing phase based on the same training model. As shown in Table 7, the ensemble update scheme costs 7.639 s in the validating phase. In the testing phase, it requires an additional 34.2% time overhead to learn the weights of L_1, L_2, and L_3.

4.4.4. The effectiveness of model updating

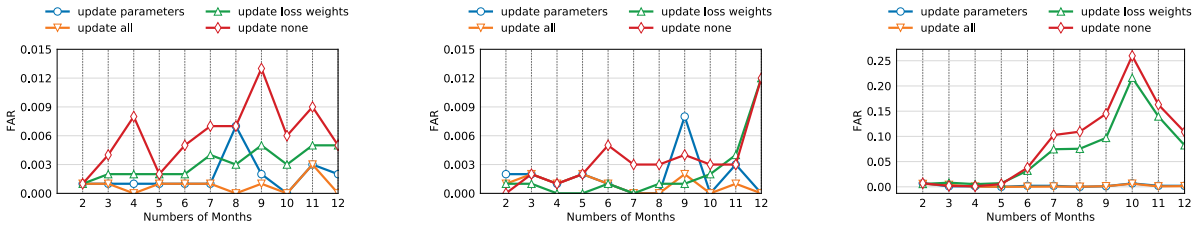
Although the necessity and effectiveness of model updating have been verified by prior work [7,14], they are all limited to supervised machine learning models. To estimate the effectiveness using parameters updating and loss weights updating, we evaluate the prediction performance of SPAE by four schemes:

- update parameters of the neural network only.



(a) STA (b) STB (c) STC

Fig. 10. FDRs under different update schemes on STA, STB, and STC.



(a) STA (b) STB (c) STC

Fig. 11. FARs under different update schemes on STA, STB, and STC.

- update weights of loss functions only.
- update both of them (used by SPAE).
- update none of them.

Fig. 11 shows the FARs of three schemes on STA, STB, and STC, respectively. It can be seen that when setting FDR around 85% (as mentioned in 4.2), we can achieve almost 0% FAR for the model updating both parameters and weights. In addition, as shown in Fig. 10, updating only the weights will show degradation in FDR in the later months, and updating only the parameters will show unstable prediction performance in some months. In contrast, updating both weights and parameters does not suffer from these problems. As a result, updating both the weights and the parameters improves FDR by 10.9% and reduces FAR by 0.018% compared to the other three update schemes on average. This phenomenon suggests that both weights and parameters suffer from the aging problem. We believe that it is necessary to continuously update the parameters of the neural network and the weights of loss functions with the latest data [7]. In addition, we find that a severe model aging problem occurs on the STC dataset without updating the model parameters.

5. Related work

To enhance the failure prediction performance, statistical techniques are proposed based on SMART attributes. Hughes et al. [46] proposed two statistical methods to improve the prediction performance. They regarded the disk failure prediction as an anomaly detection by using the Wilcoxon rank-sum test and OR-ed single variate test and achieved a 60% FDR with 0.5% FAR on a dataset composed of 3744 disks with 36 failed disks. Also from the perspective of anomaly detection, Wang et al. [18,47] proposed using Mahalanobis distance (MD) for failure detection, which aggregated the input variables into one index and detected failed disks by setting an appropriate threshold. This method delivered a 68% FDR with 0% FAR on the same dataset used by [6], which was a small-sized and balanced dataset.

Besides statistical schemes, machine learning methods have also been employed to predict disk failures and demonstrated to outperform the former. Hamerly and Elkan [48] employed two Bayesian approaches named NBEM (Naive Bayes Expectation–Maximization) and supervised naive Bayes classifier, respectively. Both algorithms were tested on a dataset from Quantum Inc., including 1927 working hard disks and 9 failed disks, and achieved promising prediction performance. Meanwhile, the supervised naive Bayesian classifier was robust against irrelevant attributes. Murray et al. [8] constructed their failure prediction systems based on support vector machine (SVM) and unsupervised clustering respectively and compared them with two non-parametric statistical tests (rank-sum and reverse arrangements test). Experimental results showed that the rank-sum method achieved the best prediction performance with 33.2% FDR at 0.5% FAR. In their subsequent work [6], they designed a novel algorithm by combining multiple-instance learning framework and naive Bayesian classifier, the results of which showed that the SVM achieved the best performance, 50.6% FDR at 0% FAR, with all selected features. However, the rank-sum test outperformed the SVM when using certain small sets of SMART attributes with 28.1% FDR at 0% FAR. Zhu et al. [14] implemented a backward propagation (BP) neural network model and an improved SVM using SMART attributes as features. Both of these models achieved satisfactory prediction performance with low FARs and the BP neural network model obtained FDR of more than 95%.

All the methods mentioned above deal with the prediction as an offline training process, which suffers from the model aging problem. Recently, [7,15] attempted to train prediction models in online mode, and both works achieved high FDRs with low FARs. Moreover, both works demonstrated that random forests achieved superior prediction accuracy. However, these online methods have three drawbacks. First, they all require a large number of failed samples for training, limiting their applications in the initial period of model deployment. For instance, in [7], it takes 4–6 months for models to achieve acceptable prediction performance, which is called the "cold start problem" in this paper. Second, high prediction performance largely depends on complicated manual feature engineering, which is time-consuming and cost-expensive. Last but not least, they need to perform additional operations, *i.e.*, re-sampling [7] or cost-sensitive learning [15], to deal with the data imbalance issue.

In this paper, we attribute the cold start problem to data imbalance issues. Therefore, we transform the disk failure prediction problem into an anomaly detection problem and present a GAN-based semi-supervised method. According to the key idea of

anomaly detection that only healthy disks are needed for training, our proposed approach can apply to small-scale data centers and in the initial period of model deployment. The use of CNN replaces the manual feature extraction. Furthermore, our model can be fine-tuned to dynamically adapt to new patterns of data and operate without concern for the model aging problem.

6. Conclusion and future work

We propose a novel GAN-based anomaly detection and ensemble update approach for lifelong disk failure prediction, called SPAE. Unlike traditional supervised machine learning methods, SPAE uses only healthy disks for model building, thus avoiding the data imbalance issue and eliminating the cold start problem confronted by supervised counterparts. Moreover, our model is trained by leveraging CNN's powerful feature extraction ability, which captures the temporal locality contained in constructed image-like 2D-SMART attributes. In addition, we introduce ensemble learning to update the model using failed disks. We evaluate our approach using three real-world datasets. The results confirm that the proposed approach is effective and outperforms supervised counterparts in both the initial period and the long-term use of model deployment. Due to the rapid development of disk products, data centers will continue to introduce new models of disks. To ensure the lifetime failure prediction performance of SPAE, we need it to be adaptive to unknown models of disks. Recently, significant progress has been made in the field of continuous learning [49]. In the future, we consider extending SPAE on the continuous learning framework to improve the generalizability and stability of the model.

CRediT authorship contribution statement

Yu Liu: Formal analysis, Investigation, Methodology, Validation, Writing editing, Funding acquisition. **Yunchuan Guan:** Data curation, Software, Validation, Visualization, Writing editing. **Tianming Jiang:** Formal analysis, Investigation, Methodology, Writing – original draft, Funding acquisition. **Ke Zhou:** Project administration, Supervision. **Hua Wang:** Project administration, Supervision. **Guangxing Hu:** Resources. **Ji Zhang:** Resources. **Wei Fang:** Supervision. **Zhuo Cheng:** Supervision. **Ping Huang:** Writing editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Yu Liu reports financial support was provided by National Natural Science Foundation of China. Tianming JIang reports financial support was provided by China Postdoctoral Science Foundation. Tianming Jiang reports financial support was provided by Basic Scientific Research of China University. @hust.edu.cn @huawei.com.

Data availability

Data will be made available on request.

Acknowledgments

This work is also done in Key Laboratory of Information Storage System and Ministry of Education of China. It is supported by the National Natural Science Foundation of China (key program) No. 62232007, the National Natural Science Foundation of China No. 61902135, the Natural Science Foundation of Hubei Province No. 2022CFB060, the China Postdoctoral Science Foundation under Grant No. 2021M701367, the Basic Scientific Research of China University under Grant No. CCNU21XJ020 and No. CCNU22QN016, and the Ministry of Education - China Mobile Research Funding No. MCM20180406.

References

[1] S. Ghemawat, H. Gobioff, S.-T. Leung, The google file system, in: ACM SIGOPS Operating Systems Review, Vol. 37, ACM, 2003, pp. 29–43.
[2] C. Huang, H. Simitci, Y. Xu, A. Ogus, B. Calder, P. Gopalan, J. Li, S. Yekhanin, et al., Erasure coding in windows azure storage, in: Proceedings of ATC, Boston, MA, 2012, pp. 15–26.
[3] Z. Huang, H. Jiang, K. Zhou, C. Wang, Y. Zhao, XI-code: A family of practical lowest density MDS array codes of distance 4, IEEE Trans. Commun. 64 (7) (2016) 2707–2718.
[4] J. Li, R.J. Stones, G. Wang, Z. Li, X. Liu, K. Xiao, Being accurate is not enough: New metrics for disk failure prediction, in: Proceedings of SRDS, IEEE, 2016, pp. 71–80.
[5] B. Allen, Monitoring hard disks with smart, Linux J. 2004 (117) (2004) 74–77.
[6] J.F. Murray, G.F. Hughes, K. Kreutz-Delgado, Machine learning methods for predicting failures in hard drives: A multiple-instance application, J. Mach. Learn. Res. (2005) 783–816.
[7] J. Xiao, Z. Xiong, S. Wu, Y. Yi, H. Jin, K. Hu, Disk failure prediction in data centers via online learning, in: Proceedings of ICPP, ACM, 2018, pp. 1–10.
[8] J.F. Murray, G.F. Hughes, K. Kreutz-Delgado, Hard drive failure prediction using non-parametric statistical methods, in: Proceedings of ICANN/ICONIP, 2003.
[9] Y. Zhao, X. Liu, S. Gan, W. Zheng, Predicting disk failures with HMM- and HSMM-based approaches, in: Proceedings of ICDM, Springer, 2010, pp. 390–404.
[10] M. Goldszmidt, Finding soon-to-fail disks in a haystack, in: Proceedings of HotStorage, 2012.
[11] J. Li, X. Ji, Y. Jia, B. Zhu, G. Wang, Z. Li, X. Liu, Hard drive failure prediction using classification and regression trees, in: Proceedings of DSN, IEEE, 2014, pp. 383–394.
[12] W. Yang, D. Hu, Y. Liu, S. Wang, T. Jiang, Hard drive failure prediction using big data, in: Proceedings of SRDS Workshops, IEEE, 2015, pp. 13–18.
[13] M.M. Botezatu, I. Giurgiu, J. Bogojeska, D. Wiesmann, Predicting disk replacement towards reliable data centers, in: Proceedings of KDD, ACM, 2016, pp. 39–48.
[14] B. Zhu, G. Wang, X. Liu, D. Hu, S. Lin, J. Ma, Proactive drive failure prediction for large scale storage systems, in: Proceedings of MSST, IEEE, 2013, pp. 1–5.
[15] Y. Xu, K. Sui, R. Yao, H. Zhang, Q. Lin, Y. Dang, P. Li, K. Jiang, W. Zhang, J.-G. Lou, et al., Improving service availability of cloud systems by predicting disk error, in: Proceedings of ATC), 2018, pp. 481–494.
[16] K.V. Vishwanath, N. Nagappan, Characterizing cloud computing hardware reliability, in: Proceedings of SoCC, ACM, 2010, pp. 193–204.
[17] B. Schroeder, G.A. Gibson, Disk failures in the real world: What does an mttf of 1, 000, 000 hours mean to you? in: Proceedings of FAST, Vol. 7, 2007, pp. 1–16.
[18] Y. Wang, Q. Miao, E.W. Ma, K.-L. Tsui, M.G. Pecht, Online anomaly detection for hard disk drives based on Mahalanobis distance, IEEE Trans. Reliab. 62 (1) (2013) 136–145.
[19] W. Jiang, C. Hu, Y. Zhou, A. Kanevsky, Are disks the dominant contributor for storage failures?: A comprehensive study of storage subsystem failure characteristics, ACM Trans. Storage 4 (3) (2008) 7.
[20] H. He, Y. Ma, Imbalanced Learning: Foundations, Algorithms, and Applications, John Wiley and Sons, 2013.
[21] Z. Yu, G. Liu, Sliced recurrent neural networks, 2018, arXiv preprint arXiv: 1807.02291.
[22] T. Jiang, J. Zeng, K. Zhou, P. Huang, T. Yang, Lifelong disk failure prediction via GAN-based anomaly detection, in: 37th IEEE International Conference on Computer Design, ICCD 2019, Abu Dhabi, United Arab Emirates, November 17-20, 2019, IEEE, 2019, pp. 199–207.
[23] S. Ha, J.-M. Yun, S. Choi, Multi-modal convolutional neural networks for activity recognition, in: Proceedings of SMC, IEEE, 2015, pp. 3017–3022.
[24] S. Ha, S. Choi, Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors, in: Proceedings of IJCNN, IEEE, 2016, pp. 381–388.
[25] C.-L. Yang, C.-Y. Yang, Z.-X. Chen, N.-W. Lo, Multivariate time series data transformation for convolutional neural network, in: 2019 IEEE/SICE International Symposium on System Integration (SII), IEEE, 2019, pp. 188–192.
[26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Proceedings of NIPs, 2014, pp. 2672–2680.
[27] S. Akcay, A. Atapour-Abarghouei, T.P. Breckon, GANomaly: Semi-supervised anomaly detection via adversarial training, 2018, arXiv preprint arXiv: 1805.06725.

[28] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: P.L. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a Meeting Held December 3-6, 2012, Lake Tahoe, Nevada, United States, 2012, pp. 1106–1114.
[29] BACKBLAZE, The backblaze hard drive data and stats, 2018, https://www.backblaze.com/b2/hard-drive-test-data.html [Online; accessed 1-february-2018].
[30] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[31] F.T. Liu, K.M. Ting, Z. Zhou, Isolation forest, in: Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy, IEEE Computer Society, 2008, pp. 413–422.
[32] J. Fan, S. Upadhye, A. Worster, Understanding receiver operating characteristic (ROC) curves, Can. J. Emerg. Med. 8 (1) (2006) 19–20.
[33] E. Pinheiro, W.-D. Weber, L.A. Barroso, Failure trends in a large disk drive population, in: Proceedings of FAST, Vol. 7, 2007, pp. 17–23.
[34] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[35] L. Breiman, Random forests, Mach. Learn. 45 (2001) 5–32.
[36] C. Cortes, V. Vapnik, Support vector machine, Mach. Learn. 20 (3) (1995) 273–297.
[37] F.T. Liu, K.M. Ting, Z.-H. Zhou, Isolation forest, in: 2008 Eighth Ieee International Conference on Data Mining, IEEE, 2008, pp. 413–422.
[38] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S.A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification, in: International Conference on Machine Learning, PMLR, 2018, pp. 4393–4402.
[39] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 27.
[40] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[41] Y. Zhao, Z. Nasrullah, Z. Li, Pyod: A python toolbox for scalable outlier detection, J. Mach. Learn. Res. 20 (96) (2019) 1–7.
[42] H. Xu, Python Deep Outlier/Anomaly Detection (DeepOD).
[43] D. Berrar, Cross-validation, 2019.
[44] J. Zhang, K. Zhou, P. Huang, X. He, M. Xie, B. Cheng, Y. Ji, Y. Wang, Minority disk failure prediction based on transfer learning in large data centers of heterogeneous disk systems, IEEE Trans. Parallel Distrib. Syst. 31 (9) (2020) 2155–2169.
[45] P. Sollich, A. Krogh, Learning with ensembles: How overfitting can be useful, in: D.S. Touretzky, M. Mozer, M.E. Hasselmo (Eds.), Advances in Neural Information Processing Systems 8, NIPS, Denver, CO, USA, November 27-30, 1995, MIT Press, 1995, pp. 190–196.
[46] G.F. Hughes, J.F. Murray, K. Kreutz-Delgado, C. Elkan, Improved disk-drive failure warnings, IEEE Trans. Reliab. 51 (3) (2002) 350–357.
[47] Y. Wang, E.W. Ma, T.W. Chow, K.-L. Tsui, A two-step parametric method for failure prediction in hard disk drives, IEEE Trans. Ind. Inform. 10 (1) (2014) 419–430.
[48] G. Hamerly, C. Elkan, et al., Bayesian approaches to failure prediction for disk drives, in: Proceedings of ICML, Citeseer, 2001, pp. 202–209.
[49] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, T. Tuytelaars, A continual learning survey: Defying forgetting in classification tasks, IEEE Trans. Pattern Anal. Mach. Intell. 44 (7) (2021) 3366–3385.

Yu Liu received the Ph.D. degree from HUST, China, in 2017. He is an associated researcher of the School of Computer Science and Technology, HUST. His main research interests include similarity-hash-based smart storage, dark data, and AIOps for data and systems. He has more than 30 publications in journals and international conferences including TCyber, TMM, TDS, SIGMOD, DAC, IJCAI, ACM MM, ICME, ICMR, APWeb-WAIN, PR, FGCS, etc.

Yunchuan Guan is the Ph.D. candidate in WNLO, HUST, China. His main research interest includes machine learning, disk reliability and intelligent storage.

Tianming Jiang received the Ph.D. degree in WNLO, HUST, China. Currently, he is an assistant professor at School of Information Management, Central China Normal University, China. His main research interest includes machine learning, disk reliability and recommender system. He has published papers in various international conferences and journals, including DATE, ICCD, JPDC, etc.

Ke Zhou received the BE, ME, and Ph.D. degrees in computer science and technology from HUST, China, in 1996, 1999, and 2003, respectively. He is a professor of the School of Computer Science and Technology and WNLO, HUST. His main research interests include computer architecture, cloud storage, parallel I/O, and storage security. He has more than 50 publications in journals and international conferences, including TPDS, PEVA, FAST, ATC, MSST, MM, INFOCOM, SYSTOR, MASCOTS, ICC, etc. He is a member of the IEEE and a member of the USENIX.

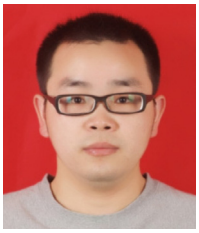
Hua Wang Hua Wang received the BE, ME, and PhD degrees in computer science and technology from HUST, China, in 1996, 2000, and 2009, respectively. She is a professor of Wuhan National Laboratory for Optoelectronics, HUST.

Guangxing Hu is the Ph.D. candidate in WNLO, HUST, China. His main research interest includes machine learning, anomaly detection and root cause analysis.

Ji Zhang is now an expert engineer in Huawei and also a postdoc in University of Amsterdam. He received his doctor degree in computer science and technology from HUST and he had been a visiting scholar in the Center for Data Science of New York University. His major is computer system structure and his research interests are storage system optimization and data management. He had published papers in international conferences and journals including ATC, SIGMOD, ICPP, DAC, VLDB, VLDBJ, TPDS and NEDB, etc.

Wei Fang received his B.Sc. and Ph.D. degrees from the University of Science and Technology of China in 2006 and 2011 respectively. Now he is a senior engineer in Huawei Technologies Co.,Ltd. His main research interests include AIOps and intelligent storage system.

Zhuo Cheng is the Eng.D candidate in Department of Computer Science and Technology, Tsinghua University. He received Master Degree in Computer's Architecture from WNLO, HUST. He is Director of Innovation Center of Data Storage, HUAWEI Technology. His main research interest includes Intelligent Storage, Hybrid Cloud Storage, Data Management, etc.



Ping Huang received the Ph.D. degree from HUST, China, in 2013. He is currently a research assistant in the Department of Computer and Information Sciences, Temple University, Philadelphia, Pennsylvania. His main research interest includes nonvolatile memory, operating system, distributed systems, DRAM, GPU, Key–value systems, etc. He has published papers in various international conferences and journals including SYSTOR, NAS, MSST, ATC, Eurosys, IFIP Performance, INFOCOM, SRDS, MASCOTS, ICCD, JSA, PEVA, Sigmetrics, ICPP, TPDS, TOS, etc.