Multi-task Hierarchical Classification for Disk Failure Prediction in Online Service Systems

Yudong Liu*, Hailan Yang*, Pu Zhao*, Minghua Ma*, Chengwu Wen*, Hongyu Zhang§, Chuan Luo*, Qingwei Lin*, Chang Yi†, Jiaojian Wang†, Chenjian Zhang†, Paul Wang†, Yingnong Dang‡, Saravan Rajmohan†, Dongmei Zhang*

*Microsoft Research, †Microsoft 365, ‡Microsoft Azure, §The University of Newcastle

ABSTRACT

One of the most common threats to online service system's reliability is disk failure. Many disk failure prediction techniques have been developed to predict failures before they actually occur, allowing proactive steps to be taken to minimize service disruption and increase service reliability. Existing approaches for disk failure prediction do not differentiate among various types of disk failure. In industrial practice, however, different product teams treat distinct types of disk failures as different prediction tasks in large-scale online service systems like Microsoft 365. For example, team A is concerned with physical disk errors, while team B focuses on I/O delay. In this paper, we propose *MTHC* (Multi-Task Hierarchical Classification) to enhance the performance of disk failure prediction for each task via multi-task learning. In addition, *MTHC* introduces a novel hierarchy-aware mechanism to deal with the data imbalance problem, which is a severe issue in the area of disk failure prediction. We show that *MTHC* can be easily utilized to enhance most state-of-the-art disk failure prediction models. Our experiments on both industrial and public datasets demonstrate that such disk failure prediction models enhanced by *MTHC* performs much better than those models working without *MTHC*. Furthermore, our experiments also present that the hierarchical-aware mechanism underlying *MTHC* can alleviate the data imbalance problem and thus improve the practical performance of various disk failure prediction models. Experiments for online industrial dataset in Microsoft 365 also demonstrates the effectiveness of our *MTHC*. [1]

CCS CONCEPTS

• **Computer systems organization** → **Cloud computing**; • **Computing methodologies** → *Neural networks.*

KEYWORDS

Disk failure prediction; Multi-task; Hierarchical classification

ACM Reference Format:

Yudong Liu*, Hailan Yang*, Pu Zhao*, Minghua Ma*, Chengwu Wen*, Hongyu Zhang§, Chuan Luo*, Qingwei Lin*, Chang Yi†, Jiaojian Wang†, Chenjian Zhang†, Paul Wang†, Yingnong Dang‡, Saravan Rajmohan†, Dongmei Zhang*. 2022. Multi-task Hierarchical Classification for Disk Failure Prediction in Online Service Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22), August 14–18, 2022, Washington, DC, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3534678.3539176

[1]Qingwei Lin is the corresponding author of this paper.

KDD '22, August 14–18, 2022, Washington, DC, USA

ACM ISBN 978-1-4503-9385-0/22/08…$15.00
https://doi.org/10.1145/3534678.3539176

1 INTRODUCTION

Online service systems, such as Facebook, Microsoft 365, and Gmail, are responsible for providing online services for millions of customers [2, 31]. Enhancing high service reliability is of great importance to improve user experience and prevent financial loss [7]. However, failures of these systems are still inevitable. Disk failures are one of the most common types of failures in online service systems [5, 19, 20, 25, 28], which could result in service interruptions and potentially financial loss. For example, disk failure accounts for 76-95 percent of all hardware failed components in data centers, according to a recent study [18]. Therefore, it is important to proactively predict disk failures.

To eliminate the loss caused by disk failures, many approaches [5, 12–14, 27, 28, 32–34, 37, 38] have been proposed to predict disk failures in advance. Current state-of-art disk failure prediction approaches based on temporal sequential models, such as LSTM [34], RNN [32], Transformer [16], and TCNN [28]. These approaches usually regard the task of disk failure prediction as a binary classification problem, and build a model to predict whether a disk will fail or not in the near future. The input of the classification model is typically SMART (Self-Monitoring, Analysis and Reporting Technology) data [1], which records status of disks and provides important indicators during the whole lifetime of disks. Engineers may take proactive actions in response to the disk failure prediction.

Existing methods generally concern a disk will fail or not in the near future, without differentiating among various disk failure types. However, our first insight is that predicting different types of disk failures is an urgent need in industrial practice. Specifically, in the large-scale Microsoft 365 (short as M365) online service systems, different teams focus on different types of disk failures. For example, team A mainly care about physical disk errors, while team B concentrate on I/O latency. Intuitively, the current solution is to build multiple disk failure prediction models for each team's specific prediction task. These independent models, on the other hand, ignore the correlation among different types of disk failure prediction. For example, disk failures caused by intensive I/O operations may also result in physical disk damage. That is, considering different tasks is pursued in practice, which motivates us to adopt multi-task learning.

In addition, since disks usually keep healthy for several months and even several years until failure, the numbers of failed disks and healthy ones are extremely imbalanced, which significantly affects failure prediction accuracy [16]. To alleviate the data imbalance, existing approaches utilize under-sampling or over-sampling methods to balance the ratio between the numbers of failed disks and healthy ones. However, these sampling methods change the data distribution and tend to cause overfitting, so they cannot achieve good performance in real scenario. Also, we gain another insight from practice that disk physical structure needs to be considered. That is, disks in the same machine share an identical computing environment, thus their status is correlated. Intuitively, when a disk fails, the status of the corresponding machine will be unhealthy. Because a machine typically contains multiple disks, the numbers of healthy and unhealthy machines are more evenly distributed than the numbers of healthy and unhealthy disks. Thus, we can leverage machine-level prediction to handle the data imbalance problem.

In this paper, to deal with the two issues discussed above, we propose a Multi-Task Hierarchical Classification (*MTHC*) framework. *MTHC* consists of two mechanisms, *i.e.*, multi-task mechanism and hierarchy-aware mechanism. Different from existing methods, which do not differentiate among various types of disk failures, our multi-task mechanism aims to capture the shared information among different types of disk failures, and thus applies multi-task learning technique. Although different tasks correspond to different types of disk failures, they can share networks to extract common feature embeddings. The intuition is that disks from different tasks are usually in the same data format, and there are non-negligible relations among different types of disk failures, which can help enhance the performance of each task. In this way, compared to dealing with each single task individually, multi-task mechanism shares common feature embeddings among tasks and could help improve the prediction accuracy of each task.

To deal with the data imbalance problem in the scenario of disk failure prediction [5, 12, 27, 28, 33, 34], we propose to conduct two-step hierarchical classification. We firstly predict whether the machine is unhealthy, *i.e.*, contains failure disks. Compared to disks, the numbers of unhealthy machines and healthy ones are not so imbalanced. If the machine is predicted as unhealthy, we then predict each disk on the machine. That is, our proposed *MTHC* can alleviate the data imbalance problem of disk failure prediction and enhance the prediction performance.

Noted that *MTHC* is orthogonal with previous models, *i.e.*, previous models can be very easily integrated into our *MTHC* framework. For the purpose of evaluating the effectiveness and robustness of *MTHC*, based on industrial dataset and two public datasets, we conduct extensive experiments to compare various disk failure prediction models enhanced by *MTHC* with those models working without *MTHC*. The experiments on three datasets indicate that *MTHC* considerably enhances the performance of disk failure prediction. Furthermore, *MTHC* have been successfully applied to

The main contributions of this paper are as follows:

- We reveal disk failure prediction challenges in industrial practice, *i.e.*, different disk failure tasks and data imbalanced problem. We design a framework *MTHC* based on domain-specific insights.
- We point out that different disk failure tasks are of high correlation and thus propose multi-task mechanism to enhance the performance of each task. Besides, we propose hierarchy-aware mechanism to deal with the data imbalance issue in the scenario of disk failure prediction.
- Extensive experiments on industrial dataset and public datasets demonstrate that our proposed *MTHC* framework considerably enhances the performance various disk failure prediction models.

2 BACKGROUND AND MOTIVATION

2.1 Disk Failure Prediction

In practice, disk failures are not only caused by hardware breakdown, but also related to heavy usage, such as intensive I/O and overheating [15, 17].

Disk failure prediction has been arousing wide concern with the rapid expansion of storage systems in data centers. With the development of deep learning, nowadays many researchers derive disk failure prediction models using deep learning techniques based on SMART data [15]. SMART is a self-monitoring system supported by most disk manufacturers, and records disk attributes such as *"Raw_Read_Error_Rate"* and *"Power_On_Hours"*, which monitor the internal health status of disks [1]. Deep learning based methods regard disk failure prediction as a binary classification problem, leveraging neural networks such as recurrent neural network (RNN) [32], long short-term memory (LSTM) [34], and temporal convolutional neural network (TCNN) [28].

2.2 Different Types of Disk Failures

Previous disk failure prediction methods only concern whether disks will fail or not in the near future. In practice, however, different teams in the M365 online service systems focus on distinct disk problems and mitigation methods, resulting in different perspectives on failure. The below are two important type of disk failures.

- **Hardware failure** [17]: Physical disk errors, such as disk aging, missing file system fields, and read-write header damage, are the primary concerns of team A.
- **Performance failure** [15]: I/O delay, which is frequently caused by overload, is more important to team B.

Therefore, in practice, especially in large-scale companies, to deal with their own disk failure problems, each team maintains a disk "failure" prediction model. Each model deals with a binary classification problem, where the class corresponds to the specific type of disk failure, such as physical disk errors and high I/O latency.

Although the disk failure prediction models from different teams deal with different failure issues, they are common in some aspects. Firstly, different teams utilize the same data format, *i.e.* SMART data (Self-Monitoring, Analysis, and Reporting Technology) [1]. Secondly, based on the same data format, different teams usually utilize similar models, *i.e.*, temporal sequence models such as LSTM to predict disk failures. Thirdly, nonnegligible correlation exists among disks with different types of disk failures, which can be

utilized [6]. For example, for both disks with physical failures and high I/O latency, the related feature values in disk data, such as read/write time might both deviate from normal interval. A disk with physical errors could usually lead to high I/O latency.

Considering the common properties mentioned above, instead of building an individual model for each team, we can regard the prediction for each team as a task and integrate tasks from different teams as a multi-task problem. Specifically, we aggregate disk data from different teams and feed the whole data into a multi-task model. The model contains several tasks, and each task deal with the specific type of disk failure from corresponding team. To utilize the feature similarity among disks with different types of disk failures, we share the embedding layers to extract common features for disks, *i.e.*, hard parameter sharing. In this way, based on multi-task learning, each task learns embedded feature from other tasks, and the performance of each individual task will be enhanced.

2.3 Hierarchical infrastructure

Previous disk failure prediction methods just deal with each disk individually, without considering the correlation among "similar" disks. However, in M365 online service systems, disks are deployed in a hierarchical infrastructure. Specifically, the top-down hierarchical structure in M365 online service systems can be formulated as: *datacenter*, *rack*, *server*, and *disk*. Disks with the same *server* field denotes that they are deployed on the same machine, and each machine contains a number of disks.

Under the circumstance, disks on the same machine share some common properties in two aspects. Firstly, for the convenience of deployment and management, disks from the same machine are usually in the same configurations. Secondly, disks on the same machine obviously share many resources, such as power and peripheral devices. These two aspects indicate that the status of disks from the same machine are more similar than those from different machines.

Since disks usually stay healthy for several months and even several years until failure, the number of failed disks and healthy ones is extremely imbalanced. Thus disk failure prediction methods have been facing the imbalance problem, which could affect the performance of prediction models. Considering the common properties mentioned above, we can alleviate the imbalance issue via the hierarchical infrastructure in M365 online service systems. As illustrated before, disks from the same machine have strong correlations. Instead of predicting the failure of each disk individually, we firstly predict whether the machine is unhealthy. We continually predict each disk on the machine if so and stop if not. To this end, we regard disk failure prediction as a hierarchical classification problem by reducing the data imbalance, which can enhance the performance of disk failure prediction.

3 METHOD

3.1 Problem Definition

In our proposed model, we formulate disk failure prediction into a multi-task hierarchical classification problem. In practice, each team focuses on one specific type of disk failure, which is regarded as a task. The total number of tasks is denoted as K. Generally, disks are deployed in physical machines. There are N machines in total and each machine contains D disks. For each disk, a feature vector of n attributes of a disk's status is recorded at each timestamp. Specifically, $X_{i,j}$ denotes the feature of j^{th} disk on the i^{th} machine, which records feature vector from timestamp t_i to timestamp $t_i + h$ (t_i is the beginning timestamp). Therefore, the shape of $X_{i,j}$ is $h \times n$. The training set consists of pairs in the format of $\{X_{i,j}, Y_{i,j}\}$. As mentioned above, $X_{i,j}$ denotes the time series feature of corresponding disk. $Y_{i,j}$ is a one-hot vector whose length equals to the number of tasks. $Y_{i,j,k} = 1$ means the disk will encounter with the k^{th} type of disk failure. Our goal is to maximize the accuracy of estimating whether a disk will fail or not for each task.

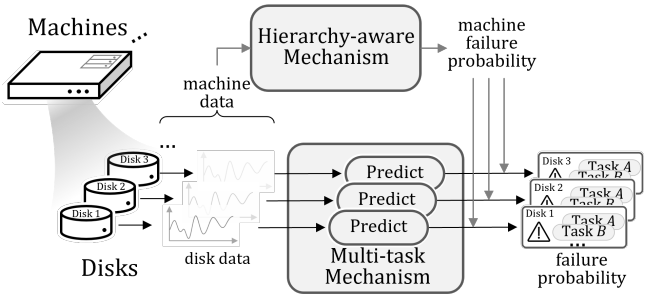

Figure 1: MTHC framework

3.2 Overview

As introduced before, *MTHC* enhances the performance of disk failure prediction from two aspects: multi-task and hierarchical classification. In this section, we briefly overview two mechanisms of our method: multi-task mechanism and hierarchy-aware mechanism. The framework of *MTHC* is shown as Figure 1.

- **Multi-task mechanism for Various Types of Disk Failures:** the input of multi-task mechanism consists of disk features from different tasks. This mechanism utilizes a shared time series model to encode the disk features from different tasks. Based on the shared encoding vectors, multi-task mechanism outputs a binary classification for each task.
- **Hierarchy-aware mechanism for Imbalanced Data:** the input of this mechanism consists features of disks on target machine. Instead of directly classifying the state of each disk, hierarchy-aware mechanism classifies whether the machine contains failure disk(s). It has no need to assess each disk if the mechanism determines that the machine is "healthy", and disks on the machine will only be reviewed if the mechanism determines that the machine includes failure disk(s).

Following two mechanisms, to aggregate two prediction results from multi-task mechanism and hierarchy-aware mechanism, we multiply them as the final prediction score.

3.3 Multi-Task mechanism

Previous disk failure prediction methods do not consider specific types of disk failures. However, in real practice, different teams in M365 online service systems focus on different types of disk

failures. For example, team A mainly cares about physical disk errors, while team B cares more about I/O latency. Generally, each team maintains their own disk failure prediction model. That is to say, the problem is regarded as several single tasks, and there is no interactions between these tasks. However, as illustrated in section 2, disks from different teams are in the same data format, *i.e.*, SMART data, and nonnegligible correlation exists among disks with different types of disk failures. Considering these properties, *MTHC* integrates different tasks together and predict failure for types of disk failures through one model, which performs as a multi-task failure prediction model. In this way, the performance of each single task will be enhanced via multi-task learning.

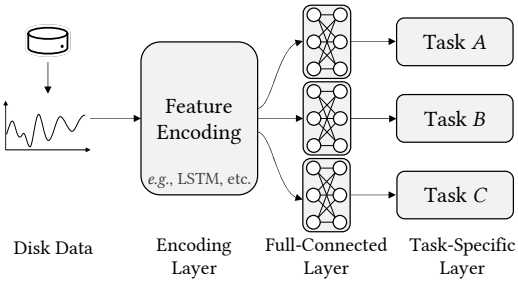

Figure 2: Multi-task mechanism

The architecture of multi-task mechanism is shown in Figure 2. The encoding layer is usually a time series model, which extracts shared feature for disks from different types of disk failures. The state-of-the-art disk failure prediction models can be very easily integrated into most multi-task mechanism. Taking LSTM as an example, a raw disk feature with the shape of $h * n$ is fed into a bidirectional LSTM network to capture the sequence information [21]. The hidden states at all time steps of LSTM are aggregated as the final encoding vector. Based on sharing encoding vector, task-aware binary classifications are implemented via full-connected layers. For each type of disk failure, a full-connected layer outputs a 2-dimension feature, followed by a softmax layer as the prediction result. For disk $X_{i,j}$, the label for the k^{th} task branch is exactly $Y_{i,j,k}$. During inference phrase, each task only focuses on the corresponding prediction result of multi-task mechanism, regardless of results from other task branches.

3.4 Hierarchy-aware mechanism

Almost all the existing approaches neglect the physical location information among disks. However, disks are deployed in physical machines, and each machine contains a number of disks. Disks deployed on the same machine are usually in the same configurations and share various resources, such as CPU, power, and so on. Therefore, these disks are highly related in their features.

If we aggregate features from all the disks from the same machine, the aggregated features could act on behalf of the state of the whole machine. Utilizing machine-level information, we can implement hierarchical classification and alleviate the imbalance classification problem of disk failure prediction. Specifically, via the aggregated features, we can firstly predict whether the whole machine is unhealthy. A machine is unhealthy if there is at least one

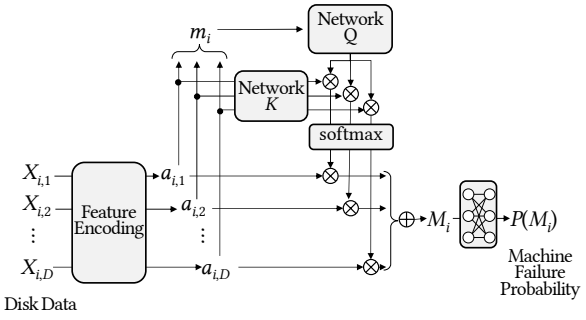

Figure 3: Hierarchy-aware mechanism

disk on the machine will fail in the near future. When the machine is considered healthy, there is no need to predict each disk on the machine. And only when the machine is predicted as unhealthy will we check each disk. The label for i^{th} machine can be calculated as:

$$Y_{m_i} = \begin{cases} 1, & \sum_{z=1}^{D} Y_{i,z} > 0 \\ 0, & \sum_{z=1}^{D} Y_{i,z} < 0 \end{cases}$$

where D denotes the number of disks on the machine. Obviously, the whole classification process is hierarchical. Considering that there exists huge imbalance between positive samples and negative samples, hierarchical classification could alleviate the issue and reduce the classification difficulties.

To obtain machine-level features, a hierarchy-aware mechanism combines all the disk features on the machine. However, instead of treating each disk equally, it is expected to focus more on those failure disks. To solve the issue, we utilize attention mechanism [11] to capture the importance of each disk. The final hierarchy-aware feature is a weighted accumulation of disk features. The process is shown in Figure 3. On the i^{th} machine, encoded vector $a_{i,j}$ for disk j is obtained via sharing encoding layer. m_i is a combination of all the $a_{i,j}$. The query vector is $q_i = Q(m_i)$, and the key vector for j^{th} disk is $k_{i,j} = K(x_{i,j})$, [30], where Q and K are fully connected networks. The weight $w_{i,j}$ for j^{th} disk is calculated as:

$$w_{i,j} = \frac{\exp(q_i \cdot k_{i,j})}{\sum_{z=1}^{D} \exp(q_i \cdot k_{i,z})} \tag{1}$$

The weighted accumulation of disk features for i^{th} machine can be calculated as:

$$M_i = \sum_{z=1}^{D} (w_{i,z} \cdot x_{i,z}) \tag{2}$$

Following the attention mechanism, the calculated hierarchy-aware feature M_i is fed into a fully connected layer and softmax layer to output the prediction of the machine, which is similar to the one in multi-task mechanism. The label for the machine is 0 if all the disks on the i^{th} machine are healthy. The label is 1 if at least one failure disk exists, regardless of the kind of disk failure.

3.5 Loss Function

In this section, we illustrate the loss function of *MTHC*. There are two loss functions in our model: multi-task loss and hierarchy-aware loss.

3.5.1 Multi-task loss. The total multi-task loss is calculated as:

$$\mathcal{L}_d = \frac{1}{K}\sum_{k=1}^{K}\mathcal{L}_{d_k} \tag{3}$$

where K denotes the number of tasks. For k^{th} task, the task-specific loss is calculated as:

$$\mathcal{L}_{d_k} = -\frac{\sum_{i,j} CE(Y_{i,j,k}, P_{d_{i,j,k}}) \cdot Y_{m_i}}{D \cdot N} \tag{4}$$

where N denotes the number of machines, D denotes the number of disks per machine, $P_{d_{i,j,k}}$ denotes the failure prediction probability of j^{th} disk on i^{th} machine for k^{th} task, and $Y_{i,j,k}$ denotes the corresponding ground truth. The function CE denotes the cross entropy loss between prediction score and ground truth, which is calculated as:

$$CE(Y, P) = Y \cdot \log(P) + (1 - Y) \cdot \log(1 - P) \tag{5}$$

Recall that Y_{m_i} represents the label for i^{th} machine, and we multiply the loss by Y_{m_i} since we only train disks when the corresponding machine indeed contains failure disks.

3.5.2 Hierarchy-aware loss. The hierarchy-aware loss is below:

$$\mathcal{L}_m = -\frac{1}{N}\sum_{i=1}^{N} CE(Y_{m_i}, P_{m_i}) \tag{6}$$

where P_{m_i} denotes the failure prediction probability of the i^{th} machine, and Y_{m_i} denotes the corresponding ground truth. Recall that the function CE denotes the cross entropy loss defined in Eq.5.

The total loss is a combination of multi-task loss and hierarchy-aware loss, which can be calculated as:

$$\mathcal{L} = \mathcal{L}_d + \mathcal{L}_m \tag{7}$$

It is noted that, since hierarchy-aware information did not attract much attention in previous methods, public datasets did not contain hierarchy-aware labels. For these datasets, our model cannot implement hierarchical classification, and our loss function only includes multi-task loss in this situation.

4 EXPERIMENTS

To evaluate the effectiveness and efficiency of *MTHC*, we conduct extensive experiments on public datasets and online industrial datasets. In this section, we first describe the experimental settings. Then, we introduce an industrial dataset and two public datasets. The experiments results are presented in Sec.4.3.

4.1 Experimental Settings

To demonstrate the robustness of *MTHC*, we adopt *MTHC* on four state-of-the-art methods which are widely used in the context of disk failure prediction: *RNN* [32], *LSTM* [34], *Transformer* [16], and *TCNN* [28]. All experiments are conducted on a workstation equipped with NVIDIA Tesla P100 GPU and CUDA 10.2. The code is implemented based on PyTorch 1.8. During the training process, we utilize Adam optimizer and set the initial learning rate as 2e-3. In addition, the training epoch is set to 100 and the batch size is 64. Noted that the public datasets do not contain machine-level labels, therefore, we do not conduct hierarchical classification on them.

Table 1: Summary of the *backblaze* dataset.

Dataset	Task 1		Task 2		Task 3	
	#Pos	#Neg	#Pos	#Neg	#Pos	#Neg
Training Set	70	6,998	531	45,176	125	12,512
Test Set	29	2,999	108	19,361	52	5,363

Table 2: Summary of the *Ali* dataset.

Dataset	Task 1		Task 2		Task 3	
	#Pos	#Neg	#Pos	#Neg	#Pos	#Neg
Training Set	56	9,908	96	9,868	64	9,900
Test Set	85	8,505	65	8,525	43	8,547

4.2 Datasets

4.2.1 Industrial Dataset. In Microsoft 365 online service system, several teams focus on disk problems. Team A focuses on physical disk errors, such as aging of disks, missing fields of file systems and damage to read-write headers. While team B cares more about I/O latency, which is usually caused by overload. We regard disk failure prediction for each team as a task. Different from public datasets which do not contain machine-level information, disks in Microsoft 365 online service systems are deployed on machines, and each machine contains a number of disks. Disk data is in the format of SMART and the attributes are collected hourly. Note that the machine-level information is included in our dataset. We predict the failure status of disks based on the latest 72-hour data. We collect online data from two teams from June 2021 to December 2021, which contains millions of disks. We treat the first five-month data as the training set and the last month as the test set.

4.2.2 Public Datasets.

Backblaze Dataset: Backblaze[2] [3] takes a snapshot of each operational hard drive, which includes basic drive information in the format of SMART statistics. It contains three-year data (October 2018 to June 2021), where each feature vector contains timestamp, disk ID, Vendor ID, SMART attributes. We divide the whole data by the *model* field that denotes different manufacturers, and regard disk failure prediction for each manufacturer as one task. We expect to enhance the failure prediction accuracy for each type of manufacturer via multi-task learning. We treat the data from October 2018 to September 2019 as the training set and the data from October 2019 to June 2021 as the test set. More detailed information for each dataset is presented in Table 1.

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

Table 3: Comparative results of various disk failure prediction models with and without *MTHC* on industry data. Note that *single* denotes single-task, *multi* denotes using multi-task mechanism, and *hierarchy* denotes using hierarchy-aware mechanism.

Approach	*Task-1*			*Task-2*		
	Precision	Recall	F1-score	Precision	Recall	F1-score
single-LSTM[34]	69.59%	65.37%	67.41%	65.13%	58.62%	61.71%
multi-LSTM	66.53%	70.56%	68.49%	67.84%	59.66%	63.49%
single-hierarchy-LSTM	74.29%	67.53%	70.75%	70.90%	59.80%	64.88%
multi-hierarchy-LSTM	75.71%	68.83%	**72.11%**	75.44%	59.31%	**66.41%**
single-RNN[32]	62.93%	70.56%	66.53%	62.66%	50.34%	55.83%
multi-RNN	63.57%	71.00%	67.08%	71.23%	52.07%	60.16%
single-hierarchy-RNN	67.87%	73.16%	70.42%	61.74%	56.21%	58.84%
multi-hierarchy-RNN	72.00%	70.13%	**71.05%**	70.04%	54.83%	**61.51%**
single-Trans[16]	63.77%	73.16%	68.15%	63.39%	55.52%	59.19%
multi-Trans	64.31%	74.89%	69.20%	66.12%	55.17%	60.15%
single-hierarchy-Trans	65.67%	76.19 %	70.54%	62.37%	61.72%	62.05%
multi-hierarchy-Trans	67.57%	75.76%	**71.43%**	67.97%	60.00%	**63.74%**
single-TCNN[28]	68.42%	67.53%	67.97%	59.41%	55.52%	57.40%
multi-TCNN	69.78%	67.97%	68.86%	58.42%	61.03%	59.70%
single-hierarchy-TCNN	65.87%	71.86%	68.84%	61.31%	57.93%	59.57%
multi-hierarchy-TCNN	70.80%	69.26%	**70.02%**	58.02%	64.83%	**61.24%**

Ali Dataset: we adopt the publicly available *Ali*[3] dataset, a real industrial data collected by the Alibaba Cloud's data centers and widely used to evaluate the performance of methods for disk failure prediction. The public Ali data contains the timestamp, serial number, disk manufacturer, disk model, normalized SMART attributes, raw SMART attributes and fault type of each disk. We randomly select three tasks according to the fault type, all of which are to predict the specific type of disk failure. We adopt the dataset from July 2017 to February 2018 as the training set, and treat March 2018 to July 2018 as the test set. More detailed information for the public dataset is listed in Table 2.

The two datasets above are both collected on a daily basis. We use consecutive 30 days of SMART data of each disk following the standard practice[4]. Specifically, we remove all-empty features and single-valued features from the raw SMART attributes, and take the rest as the final representation of disk.

4.3 Experimental Results

4.3.1 Industrial Dataset. We implement *MTHC* on Microsoft 365 online service systems, and Table 3 presents the comparative results of *MTHC* with its corresponding single model of state-of-the-art competitors on the dataset. To explore the effectiveness of two mechanisms of *MTHC*, we conduct experiments for each mechanism.

Based on only multi-task mechanism, four approaches all outperforms ones with single task. Specifically, for task 1 (physical disk errors), multi-task mechanism exceeds single task by an average of 0.89 percent of F1 score, and 2.34 percent of F1 score for task 2 (I/O latency). The result demonstrates the effectiveness and robustness of multi-task mechanism. Besides, we notice that task 2 (I/O latency) obtains much more enhancement than task 1 (physical disk errors). Because disks with physical disk errors could usually lead to high I/O latency, which help improve the performance of task 2. In contrast, disks with high I/O latency do not always indicates physical errors, which are usually caused by overload.

Based on only hierarchy-aware mechanism, four approaches exceed ones with single task by a large margin. Specifically, for task 1 (physical disk errors), the hierarchy-aware mechanism exceeds the single task by an average of 2.62 percent of F1 score, and 2.80 percent of F1 score for task 2 (I/O latency). The result demonstrates that hierarchy-aware mechanism reduces the imbalance problem of positive and negative samples and enhances the performance of disk failure prediction via hierarchical classification. In addition, compared to multi-task mechanism, hierarchy-aware mechanism obtains comparable improvement for task 1 (physical disk errors) and task 2 (I/O latency). Because disk infrastructures from two teams are very isomorphic, *i.e.*, hierarchical, and the machines from both teams contain a number of disks.

Combining multi-task mechanism and hierarchy-aware mechanism, the performance of disk failure prediction is further enhanced. Specifically, for task 1 (physical disk errors), *MTHC* Module exceeds single task by an average of 3.64 percent of F1 score, and 4.69 percent of F1 score for task 2 (I/O latency). Noted that the F1-score of online disk failure prediction is usually low, and it is not easy to

[3]https://tianchi.aliyun.com/dataset/dataDetail?dataId=70251
[4]https://tianchi.aliyun.com/competition/entrance/231775/information?lang=en-us
[5]https://tianchi.aliyun.com/competition/entrance/231775/rankingList/1

Table 4: Comparative results of various disk failure prediction models with single-task and multi-task on *BackBlaze Dataset*.

Approach	*Task-1*			*Task-2*			*Task-3*		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
single-LSTM [34]	88.00%	75.86%	81.48%	82.69%	79.63%	81.13%	71.43%	67.31%	69.31%
multi-LSTM	88.46%	79.31%	**83.64%**	90.11%	75.93%	**82.41%**	80.00%	69.23%	**74.23%**
single-RNN [32]	82.61%	65.52%	73.08%	92.31%	77.78%	84.42%	69.31%	60.31%	67.31%
multi-RNN	91.30%	72.41%	**80.77%**	94.31%	76.85%	**84.69%**	77.78%	67.31%	**72.16%**
single-Trans [16]	95.24%	68.97%	80.00%	89.36%	77.78%	83.17%	75.56%	65.38%	70.10%
multi-Trans	95.46%	72.41%	**82.35%**	88.54%	78.70%	**83.33%**	86.84%	63.46%	**73.33%**
single-TCNN [28]	81.82%	62.07%	70.59%	91.95%	74.07%	82.05%	68.89%	59.62%	63.92%
multi-TCNN	80.00%	68.97%	**74.07%**	93.27%	76.85%	**84.26%**	76.32%	55.77%	**64.44%**

Table 5: Comparative results of various disk failure prediction models with single-task and multi-task on *Ali Dataset*. Note that due to the difficulty of all tasks on this dataset, the prediction results in this dataset are low. The first on the leaderboard only achieves 49.07% F1-score[5].

Approach	*Task-1*			*Task-2*			*Task-3*		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
single-LSTM[34]	39.80%	45.88%	42.62%	42.11%	61.54%	50.00%	40.00%	46.51%	43.01%
multi-LSTM	38.57%	63.53%	**48.00%**	37.76%	83.53%	**51.92%**	45.24%	44.19%	**44.71%**
single-RNN[32]	39.17%	55.29%	45.85%	38.81%	80.00%	52.26%	39.34%	55.81%	46.15%
multi-RNN	37.59%	58.82%	**45.87%**	39.53%	78.46%	**52.58%**	48.94%	53.49%	**51.11%**
single-Trans[16]	35.77%	51.76%	42.31%	42.48%	73.85%	53.93%	50.00%	46.51%	48.19%
multi-Trans	40.68%	56.47%	**47.29%**	41.67%	76.92%	**54.05%**	46.30%	58.14%	**51.55%**
single-TCNN[28]	42.45%	52.94%	47.12%	34.65%	67.69%	45.83%	48.89%	51.16%	50.00%
multi-TCNN	46.08%	55.29%	**50.27%**	35.16%	69.23%	**46.63%**	60.00%	48.84%	**53.85%**

improve it due to severely missing data and extreme data imbalance problem in practice [16].

In all, *MTHC* based approaches significantly outperforms corresponding single-model ones, which demonstrates the effectiveness and robustness of our *MTHC*.

4.3.2 Public Datasets. We also use the public datasets (the Backblaze and Ali datasets) to evaluate the performance of the proposed approach. Note that we only evaluate the multi-task mechanism here due to the missing of machine-level information in the datasets.

The results on the Backblaze dataset are presented in Table 4. All the *multi-task* based approaches outperform corresponding single-task ones. On average, *multi-task* based approaches achieves the F1-score values with 80.21%, 83.67% and 71.04%, which are 3.92%, 0.98% and 3.38% greater than corresponding single-task approach for each task. In addition, according to Table 4, Task 1 and Task 3 contain fewer positive samples than the Task 2, while the performance enhancement of Task 1 and Task 3 is higher than that of Task 2. This demonstrates that multi-task architecture is very suitable for the failure prediction task with few positive samples.

The results on the Alibaba Cloud's dataset are shown in Table 5. The four *multi-task* based approaches exceed corresponding single-task ones by a large margin. On average, *multi-task* based approaches achieves the F1-score values with 47.86%, 51.30% and 50.31%, which are 3.38%, 0.79% and 3.47% greater than corresponding single-task approach for each task. It is noted that the result of various methods for Alibaba Cloud's data are all low due to the difficulty of the dataset. The first on the leaderboard only achieves the F1-score value with 49.07%[5].

5 DEPLOYMENT AND DISCUSSION

In this section, we introduce the deployment of our *MTHC* and lessons learned. We have run our *MTHC* pipeline on M365 large scale online service systems, which contains millions of disks to serve a huge number of customers. The whole pipeline is run on *Azure databrick*, which consists of three phases: data fetching phase, data processing phase and model processing phase.

5.1 Deployment

Data fetching phase: In this phase, service *Smart_ctl* is called hourly to collect SMART data from each server in two teams. The

collected SMART data is stored in *Azure*, and is transferred by a distributed and reliable streaming data moving tool to ensure the consistence and completeness of data.

Data processing phase: This phase contains data cleaning and feature engineering. Data cleaning is firstly scheduled to deal with missing data. Then feature engineering jobs are scheduled to obtain disk feature vectors [26, 29], which are the input of *MTHC*. Feature engineering jobs contain several operations, such as filtering irrelevant attributes, concatenating different type of attributes, and appending machine-level information for disks. We accelerate the feature engineering process via *Spark*.

Model processing phase: In this phase, we utilize disk feature vectors obtained in data processing phase and train the disk failure prediction tasks with *MTHC* LSTM model. The trained model gives the failure scores for each disk and each task. Note that each task only focuses on the corresponding failure type. Disks with high failure probability are stored in *Azure Table*. Engineers from each team query half an hour and select disks with corresponding high failure probability from Azure Table and take proactive actions. For example, team A could select disks with high physical error probability and transfer the data on the disks to healthy ones, *i.e.*, live migration.

To evaluate the effectiveness of *MTHC* from the business impact point of view, we employ A/B testing to measure the number of virtual machine interruptions saved through proactive mitigation based on failure prediction signals. According to the testing result, compared to single-task based approach, our proposed *MTHC* notably reduced the number of virtual machine interruptions per month for M365 online service systems, which was of great enhancement for the service reliability of M365 online service systems and brought considerable benefits.

5.2 Lessons Learned

During the deployment of *MTHC*, we found that missing data is a serious problem that can lead to poor prediction model performance. Missing data is frequently caused by poor data collection and transmission quality. Besides, when a disk is unhealthy, it may fail to provide SMART data, which might signal a failure due to missing data. In other words, we can utilize data missing to predict failures. However, we currently just populate zero based on the standard processing of disk failure prediction. In the future, we aim to handle the missing data systematically.

6 RELATED WORK

6.1 Disk failure prediction

In recent years, as disk failures have received increasing attention, many methods of disk failure prediction have been proposed. These methods can be roughly divided into two categories: traditional machine learning based and deep learning based.

To improve the prediction performance, traditional machine learning-based models such as support vector machines [34] and tree-based machine learning models [5, 12, 13, 27, 33] utilize SMART data for disk failure prediction. However, these traditional machine learning based approaches can't handle the temporal information well [28], while deep learning based approaches can make better use of the temporal information. Deep learning based approaches, including recurrent neural network (RNN) [32], long short-term memory (LSTM) [34] and temporal convolutional neural network (TCNN) [28], perform better than traditional machine learning based ones for disk failure prediction.

Compared to existing disk failure prediction approaches which do not focus on specific failure types, our proposed *MTHC* framework integrates disks from each team and utilize multi-task learning to enhance the performance for each single task and introduces a novel node hierarchical classification approach to deal with the extreme data imbalance of disk failure prediction problem. Note that our proposed *MTHC* is orthogonal with previous models, *i.e.*, it can be very easily to integrated previous models into *MTHC* and enhances the performance of disk failure prediction.

6.2 Multi-task learning

Multi-task learning has been used successfully across many applications of machine learning, from natural language processing [8] and speech recognition [10] to computer vision [23] and drug discovery [22, 24]. And there are many opportunities for multi-task learning on real-world problems [6]. In areas of multi-task learning, shared representations are utilized to explore common patterns among a collection of related tasks. Compared to training the models separately, these shared representations can help improve learning efficiency and prediction accuracy for the task-specific models. [6].

In the context of deep learning, the methods of multi-task learning [24, 35, 36] can be divided into two groups: soft parameter sharing and hard parameter sharing. In soft parameter sharing, each task maintains its own model with its own parameters, and the distance between the model parameters of different tasks regularized in order to encourage the parameters from different tasks to be similar [9]. Compared to soft parameter sharing, hard parameter sharing methods share the hidden layers (*i.e.*, representations) among tasks while keeping task-specific output layers for each task. And the loss function for hard parameter sharing is typically a combination of multiple loss functions corresponding to multiple tasks. Compared to soft parameter sharing, hard parameter sharing greatly reduces the risk of overfitting [4], and our *MTHC* utilizes hard parameter sharing.

7 CONCLUSION

Disk failure is one of the most frequently failing components in online service systems, which could reduce the service reliability. Disk failure prediction has been attracting extensive attention. In this paper, we propose a Multi-Task Hierarchy Classification framework named *MTHC* for disk failure prediction, which consists of two main mechanisms, *i.e.*, multi-task mechanism and hierarchy-aware mechanism. Compared to existing approaches which do not focus on specific failure types, we emphasize that different teams focuses on different types of disk failures in real practice. Multi-task mechanism integrates disks from each team and utilize multi-task learning to enhance the performance for each single task. Moreover, hierarchy-aware mechanism introduces a novel node hierarchical classification approach to deal with the extreme data imbalance of disk failure prediction problem. Our experiments on both industrial and public datasets demonstrate that multi-task based approaches achieve much better performance than corresponding single-task

based ones. Further, experiments also demonstrate that hierarchy-aware mechanism can alleviate the imbalance problem and enhance the performance of various disk failure prediction approaches.

REFERENCES

[1] Bruce Allen. 2004. Monitoring Hard Disks with SMART. *Linux Journal*.

[2] Danilo Ardagna, Barbara Panicucci, and Mauro Passacantando. 2011. A Game Theoretic Formulation of the Service Provisioning Problem in Cloud Systems. In *Proceedings of WWW*. 177–186.

[3] Backblaze. 2019. The Backblzae Hard Drive Data and Stats. https://www.backblaze.com/b2/hard-drive-test-data.html.

[4] Jonathan Baxter. 1997. A Bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning* 28, 1, 7–39.

[5] Mirela Madalina Botezatu, Ioana Giurgiu, Jasmina Bogojeska, and Dorothea Wiesmann. 2016. Predicting Disk Replacement towards Reliable Data Centers. In *Proceedings of KDD*. 39–48.

[6] Rich Caruana. 1997. Multitask learning. *Machine learning* 28, 1, 41–75.

[7] Yujun Chen, Xian Yang, Qingwei Lin, Hongyu Zhang, Feng Gao, Zhangwei Xu, Yingnong Dang, Dongmei Zhang, Hang Dong, Yong Xu, Hao Li, and Yu Kang. 2019. Outage Prediction and Diagnosis for Cloud Service Systems. In *Proceedings of WWW*. 2659–2665.

[8] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*. 160–167.

[9] Michael Crawshaw. 2020. Multi-Task Learning with Deep Neural Networks: A Survey. *ArXiv* abs/2009.09796.

[10] Li Deng, Geoffrey Hinton, and Brian Kingsbury. 2013. New types of deep neural network learning for speech recognition and related applications: An overview. In *Proceedings of ICASSP*. IEEE, 8599–8603.

[11] Chenyou Fan, Yuze Zhang, Yi Pan, Xiaoyue Li, Chi Zhang, Rong Yuan, Di Wu, Wensheng Wang, Jian Pei, and Heng Huang. 2019. Multi-Horizon Time Series Forecasting with Temporal Attention Learning. In *Proceedings of KDD*. 2527–2535.

[12] Xiaohong Huang. 2017. *Hard Drive Failure Prediction for Large Scale Storage System*. Ph. D. Dissertation. UCLA.

[13] Jing Li, Xinpu Ji, Yuhan Jia, Bingpeng Zhu, Gang Wang, Zhongwei Li, and Xiaoguang Liu. 2014. Hard Drive Failure Prediction Using Classification and Regression Trees. In *Proceedings of DSN*. 383–394.

[14] Sidi Lu, Bing Luo, Tirthak Patel, Yongtao Yao, Devesh Tiwari, and Weisong Shi. 2020. Making Disk Failure Predictions {SMARTer}!. In *Proceedings of FAST*. 151–167.

[15] Sidi Lu, Bing Luo, Tirthak Patel, Yongtao Yao, Devesh Tiwari, and Weisong Shi. 2020. Making Disk Failure Predictions SMARTer!. In *Proceedings of FAST*. USENIX Association, 151–167.

[16] Chuan Luo, Pu Zhao, Bo Qiao, Youjiang Wu, Hongyu Zhang, Wei Wu, Weihai Lu, Yingnong Dang, Saravanakumar Rajmohan, Qingwei Lin, and Dongmei Zhang. 2021. NTAM: Neighborhood-Temporal Attention Model for Disk Failure Prediction in Cloud Platforms. In *Proceedings of WWW*. 1181–1191.

[17] Ao Ma, Fred Douglis, Guanlin Lu, Darren Sawyer, Surendar Chandra, and Windsor W. Hsu. 2015. RAIDShield: Characterizing, Monitoring, and Proactively Protecting Against Disk Failures. In *Proceedings of FAST*. USENIX Association, 241–256.

[18] Ioannis Manousakis, Sriram Sankar, Gregg McKnight, Thu D. Nguyen, and Ricardo Bianchini. 2016. Environmental Conditions and Disk Reliability in Free-cooled Datacenters. In *Proceedings of ATC*. USENIX Association.

[19] Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu. 2015. A Large-Scale Study of Flash Memory Failures in the Field. In *Proceedings of SIGMETRICS*. 177–190.

[20] Molly S. Quinn, Katherine Campbell, and Mark T. Keane. 2019. The Expected Unexpected & Unexpected Unexpected. In *Proceedings of CogSci*. 2627–2633.

[21] Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. 2020. Compressive Transformers for Long-Range Sequence Modelling. In *Proceedings of ICLR*.

[22] Bharath Ramsundar, Steven Kearnes, Patrick Riley, Dale Webster, David Konerding, and Vijay Pande. 2015. Massively multitask networks for drug discovery.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems* 28.

[24] Sebastian Ruder. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. *CoRR* abs/1706.05098.

[25] Sriram Sankar, Mark Shaw, Kushagra Vaid, and Sudhanva Gurumurthi. 2013. Datacenter Scale Evaluation of the Impact of Temperature on Hard Disk Drive Failures. *ACM Transactions on Storage* 9, 2, 1–24.

[26] Huasong Shan, Yuan Chen, Haifeng Liu, Yunpeng Zhang, Xiao Xiao, Xiaofeng He, Min Li, and Wei Ding. 2019. ϵ-Diagnosis: Unsupervised and Real-time Diagnosis of Small-window Long-tail Latency in Large-scale Microservice Platforms. In *Proceedings of WWW*. 3215–3222.

[27] Jing Shen, Jian Wan, Se-Jung Lim, and Lifeng Yu. 2018. Random-Forest-Based Failure Prediction for Hard Disk Drives. *International Journal of Distributed Sensor Networks* 14, 11.

[28] Xiaoyi Sun, Krishnendu Chakrabarty, Ruirui Huang, Yiquan Chen, Bing Zhao, Hai Cao, Yinhe Han, Xiaoyao Liang, and Li Jiang. 2019. System-Level Hardware Failure Prediction using Deep Learning. In *Proceedings of DAC*. 20.

[29] Amoghavarsha Suresh and Anshul Gandhi. 2019. Using Variability as a Guiding Principle to Reduce Latency in Web Applications via OS Profiling. In *Proceedings of WWW*. 1759–1770.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of NIPS*. 5998–6008.

[31] Changjun Wang, Weidong Ma, Tao Qin, Xujin Chen, Xiaodong Hu, and Tie-Yan Liu. 2015. Selling Reserved Instances in Cloud Computing. In *Proceedings of IJCAI*. 224–231.

[32] Chang Xu, Gang Wang, Xiaoguang Liu, Dongdong Guo, and Tie-Yan Liu. 2016. Health Status Assessment and Failure Prediction for Hard Drives with Recurrent Neural Networks. *IEEE Transactions on Computers* 65, 11, 3502–3508.

[33] 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. 2018. Improving Service Availability of Cloud Systems by Predicting Disk Error. In *Proceedings of ATC*. USENIX Association, 481–494.

[34] Jianguo Zhang, Ji Wang, Lifang He, Zhao Li, and Philip S. Yu. 2018. Layer-wise Perturbation-Based Adversarial Training for Hard Drive Health Degree Prediction. In *Proceedings of ICDM*. 1428–1433.

[35] Yu Zhang. 2015. Multi-Task Learning and Algorithmic Stability. In *Proceedings of AAAI*. 3181–3187.

[36] Yu Zhang and Qiang Yang. 2017. A Survey on Multi-Task Learning. *CoRR* abs/1707.08114.

[37] Ying Zhao, Xiang Liu, Siqing Gan, and Weimin Zheng. 2010. Predicting Disk Failures with HMM- and HSMM-Based Approaches. In *Proceedings of ICDM*. 390–404.

[38] Bingpeng Zhu, Gang Wang, Xiaoguang Liu, Dianming Hu, Sheng Lin, and Jingwei Ma. 2013. Proactive Drive Failure Prediction for Large Scale Storage Systems. In *Proceedings of MSST*. 1–5.