Summary of PAKDD CUP 2020: From Organizers' Perspective

Cheng He[1(✉)], Yi Liu[1], Tao Huang[1], Fan Xu[1], Jiongzhou Liu[1], Shujie Han[2], Patrick P. C. Lee[2], and Pinghui Wang[3]

[1] Alibaba Group, Hangzhou, China
hecheng.hc@alibaba-inc.com
[2] The Chinese University of Hong Kong, Hong Kong, China
[3] Xi'an Jiaotong University, Xi'an, China

Abstract. PAKDD2020 Alibaba AI Ops Competition is jointly organized by PAKDD2020, Alibaba Cloud, and Alibaba Tianchi platform. The task of the competition is to predict disk failures in large-scale cloud computing environments. We provide SMART (Self-Monitoring, Analysis and Reporting Technology) logs on a daily basis in the production environments without any preprocessing except anonymization. The SMART logs pose great challenges for analysis and disk failure prediction in real production environments, such as data noises and the extreme data imbalance property. In this paper, we first describe the competition task, practical challenges, and evaluation metrics. We then present the statistical results of the competition and summarize the key techniques adopted in the submitted solutions. Finally, we discuss the open issues and the choices of techniques regarding the online deployment.

Keywords: Disk failure prediction · Cloud computing · AI Ops

1 Introduction

In large-scale cloud computing environments, millions of hard disk drives are deployed to store and manage massive data [2]. With such large-scale modern data centers, disk failures are prevalent and account for the largest proportion among the hardware failures in cloud data centers [20]. Disk failures may lead to service performance degradation, service unavailability, or even data loss [5]. In order to provide cloud services with high availability and reliability, cloud service providers explore proactive fault tolerance approaches to predict disk failures in advance.

For more than a decade, researchers from academia and industry have made great progress in disk failure predictions. Recent studies [6,9,11,12,15–17,19,21,22,24] conduct comprehensive analysis on SMART (Self-Monitoring, Analysis, and Reporting Technology) logs [4], and they also make use of machine learning

Jointly organized by PAKDD 2020, Alibaba Cloud and Alibaba Tianchi Platform.

C. He et al. (Eds.): AI Ops 2020, CCIS 1261, pp. 130–142, 2020.
https://doi.org/10.1007/978-981-15-7749-9_13

algorithms to achieve accurate prediction results. However, as the public datasets are limited at scale in the whole community, it is difficult to apply state-of-the-art solutions directly into production environments due to the following more strict requirements for online deployment.

- **Prediction on a daily basis.** In the production environment, the monitoring system usually collects SMART logs on a daily basis. Thus, disk failure prediction is supposed to output prediction results at the same granularity. This requirement leads to more severe data imbalance problem.
- **Data noise.** Data noise is commonplace and attributed to labeling and sampling in the complicated production environments. Specifically, administrators often detect disk failures by using the self-defined expert rules, yet these rules may change over time and cannot cover the unknown disk failures, thereby leading to noise in labeling. Also, the data collection process may be interrupted by some incidents in production, which causes missing of data during data collection.

In this paper, we first describe the competition task and related challenges in Sect. 2. We then describe the details of our proposed evaluation metrics for disk failure prediction under requirements in production environments in Sect. 3. In Sect. 4, we highlight the mainstream and novel techniques adopted in the submitted solutions. We analyze the overall statistics of submitted results in Sect. 5. Finally, we discuss the open issues in practical deployment in Sect. 6.

2 Task Description

The task of PAKDD2020 Alibaba AI Ops Competition is about reliability and availability improvement in cloud computing environments, in particular, through the predictive maintenance of disk failures. The participants of the competition are required to predict disk failures in the future 30 days based on the historical SMART logs and failure tags. We provide the SMART logs of two hard disk models from the same manufacturer over the duration for more than one year. Each disk model contains more than 100 K independent hard disk drives. To the best of our knowledge, this is the largest public dataset by size for a single disk model for disk failure prediction. As this is a supervised learning task, in addition to SMART logs, we also provide labels contained in the failure tag file collected from our trouble tickets system. All the dataset and related descriptions are available at PAKDD Cup 2020 and Tianchi website: https://tianchi.aliyun.com/competition/entrance/231775/information?lang=en-us.

Table 1 shows the metadata of training and testing SMART logs of the disk models A1 and A2. The SMART logs contain 514 columns in total, including the disk serial number, manufacturer, disk model, data collecting time, and 510 columns of the SMART attributes. We denote the SMART attributes by "smart_n", where n is the ID of the SMART attribute. Each SMART attribute has a raw value and a normalized value, which are denoted by "smart_n_raw" and "smart_n_normalized", respectively. The SMART attributes are vendor-specific,

while many of these attributes have a normalized value that ranges from 1 to 253 (with higher values representing better status). The initial default normalized value of the SMART attributes is 100 but can vary across manufacturers.

Table 2 shows the metadata of failure tags. It contains five columns including the disk serial number, manufacturer, disk model, failure time, and sub-failure type. In particular, the trouble ticket system in Alibaba Cloud detects disk failures using expert rules and reports the failure time (denoted by "fault_time") and sub-failure type (denoted by "tag") when failures occur. For the sub-failure types, we anonymize the names and map them into the numbers ranging from one to five, while we set the sub-failure type as zero by default for healthy disks.

Table 1. Metadata of training and testing sets.

Column name	Type	Description
serial_number	string	disk serial number code
manufacturer	string	disk manufacturer code
model	string	disk model code
smart_n_normalized	integer	normalized value of SMART-n
smart_n_raw	integer	raw value of SMART-n
dt	string	data collecting time

Table 2. Metadata of failure tags.

Column name	Type	Description
serial_number	string	disk serial number code
manufacturer	string	disk manufacturer code
model	string	disk model code
fault_time	string	time of failure reported in the trouble ticket system
tag	integer	ID of sub-failure type, ranging from 0 to 5

The competition has three rounds: preliminaries, semi-finals, and finals. In preliminaries, we provide the dataset of two disk models from the same manufacturer including the SMART logs and failure tags. We set the training period from July 31, 2017 to July 31, 2018 and the testing period from August 1, 2018 to August 31, 2018. During preliminaries, we first open the training and testing sets of disk model A1 for the participants. The participants can leverage the dataset to select features, construct machine learning model, and optimize their machine learning models. We then open the training and testing sets of disk model A2 for participants five days before the preliminaries deadline, so as to test the generalization of their methodology designed for disk model A1 on the

new testing set (disk model A2). We select the top 150 teams with the higher prediction accuracy on disk model A2 into semi-finals.

In semi-finals, participants have the same training sets of disk model A1 and A2 with those in preliminaries, but the testing dataset is not available offline. Instead, we merge the testing data of disk models A1 and A2 together from September 1, 2018 to September 30, 2018 and put the resulting dataset into online testing environments. We require participants to submit their prediction solutions packed in docker files to online testing environments for predicting disk failures in the testing dataset (including both disk models A1 and A2). We select the top 12 teams with higher prediction accuracy into finals.

In finals, we require the 12 teams to present their solutions, including the main idea, design of machine learning models, and workflow details (e.g., feature selection and construction as well as optimization methods). We invite experts from industry and academia as our committee to score their presentations in terms of reasonability, novelty, and completeness. The final scores are comprised of two parts, i.e., 70% for prediction results and 30% for presentation.

In this competition, we summarize the following challenges in our dataset for disk failure prediction:

- **Extremely imbalanced data.** The data imbalance problem is a well-known challenge in the machine learning community [13], meaning that classifiers tend to be more biased towards the majority class. In production, when we predict disk failure on a daily basis, the data imbalance becomes more severe and the imbalance ratio of failed disks to healthy disks is less than 0.003% in our dataset. Therefore, data imbalance is a critical issue that needs to be addressed in both the competition as well as production environments.
- **Data noise.** Data noise is commonplace and attributed to many reasons, such as network failures, software malfunction/upgrades, system or server crashes, data missing, or anomaly collected data events in monitoring systems. All these events bring noise into the dataset and compromise the expected patterns of the dataset. Data noise cannot be ignored in disk failure prediction, as it can impair the prediction accuracy. How to design and apply proper techniques in dealing with the data noise problem is also a key to improving the accuracy of disk failure prediction solutions.

We encourage participants to leverage prior studies and state-of-the-art techniques to obtain domain knowledge of the SMART logs and disks during competition period in addition to our dataset and specifications. We also build a testing environment for participants to evaluate and reproduce the submitted solution, so as to guarantee the correctness of the prediction outputs.

3 Evaluation

We evaluate the prediction results of participants' submitted solutions based on three accuracy metrics, including precision, recall, and F1-score, as defined below.

- **Precision for P-window.** We define the *precision* as the fraction of actual failed disks being predicted over all (correctly and falsely) predicted failed disks. As our objective is to evaluate whether a failed disk being predicted is an actual failure within 30 days, we define the *P-window* as a fixed-size *sliding* window starting from the first time that a disk is predicted as failure, and set the length of the P-window as 30 days. Let T denote the start date and $T + k - 1$ denote the end date of the testing period (k as 30 days in our competition). Note that the P-window may slide out of the testing period. Figure 1 illustrates how we count true positive and false positive results. If the actual failure happens within the P-window (e.g., the 1st and 4th rows), we regard the failed disk as a correctly predicted one; otherwise (e.g., the 2nd and 3rd rows), we regard the disk as a falsely predicted one.
- **Recall for R-window.** We next define the *recall* as the fraction of actual failed disks being predicted over all actual failed disks. We define the *R-window* as a fixed-size window (not sliding window) from the starting date to the end date of the testing period with the length of 30 days in our case (i.e., from T to $T + k - 1$, where k is 30 days). Figure 2 shows how we count false positive, false negative, and true positive results. If a failed disk being predicted is not failed within the R-window (the 1st and 2nd rows), we regard the disk as a falsely predicted one; otherwise, we regard the failed disk as a correctly predicted one (the 4th and 5th rows). If an actual failed disk within the R-window is not predicted, we regard the failed disk as a missed one (i.e., false negative in the 3rd row).
- **F1-score.** We follow the classical definition of F1-score as $\frac{2 \times precision \times recall}{precision + recall}$. For easy comparison, we use F1-score as the participants' score in preliminaries and semi-finals.

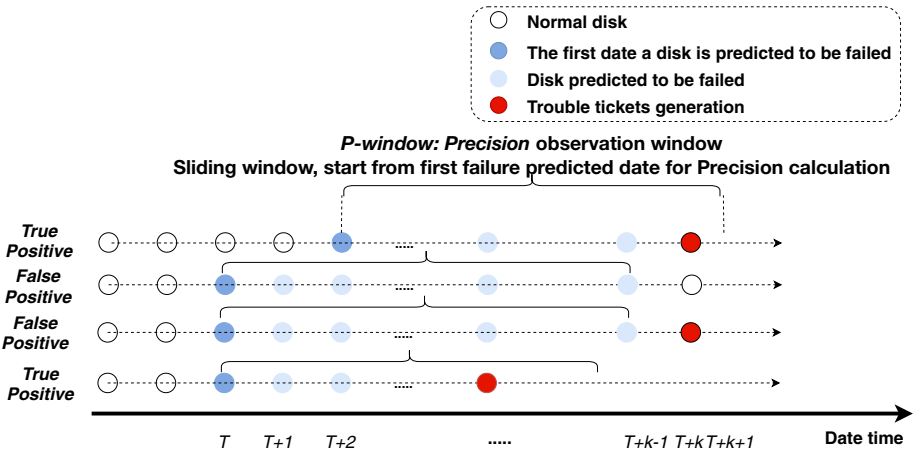


Fig. 1. Illustration of P-window.

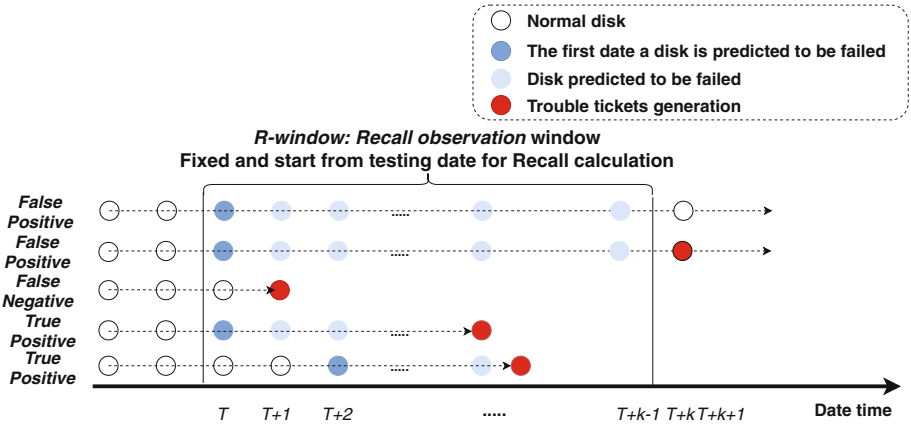
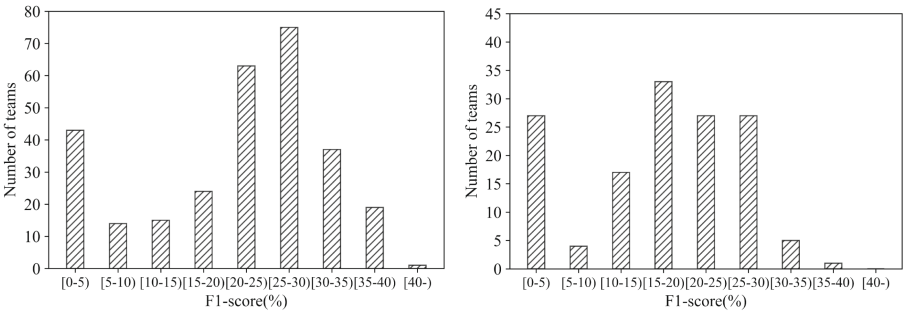


Fig. 2. Illustration of R-window.

In preliminaries, participants are required to submit prediction results in CSV format with four deterministic columns, including the manufacturer, disk model, disk serial number, and failure predicted date, based on our provided dataset. If one disk is predicted to be failed multiple times, we only take the earliest prediction date in the evaluation process and ignore the later ones. In semi-finals, we put the testing dataset on the cloud testing environments, so participants must submit their solutions packed with a docker image to predict disk failures on a daily basis. Then the auto-evaluation process gives a final score based on the aforementioned metrics. In finals, the top-ranking teams in the leaderboard are asked to present the strengths and weaknesses of their solutions. The final scores are comprised of two parts, 70% of prediction results in semi-finals and 30% of presentation results in finals.



(a) Disk model A1 (b) Disk model A2

Fig. 3. F1-score distribution of prediction results of disk models A1 and A2 in preliminaries.

4 Statistics of Submissions

In this section, we analyze the statistics of submissions on the prediction accuracy distributions, team affiliation, and the time spent of participants on the competition.

4.1 Prediction Accuracy Distributions

We first analyze the prediction accuracy distributions in preliminaries. There are 1,173 teams registered in the competition and we received 3,309 valid submissions from 291 teams for failure prediction of disk model A1. Figure 3(a) shows the distributions of F1-score for predicting failures in disk model A1. From the figure, we can see that around 19.6% teams achieve the F1-scores higher than 30%, while nearly half (47.4%) of the teams' results fall into the interval between 20% and 30% F1-score. Figure 3(b) illustrates the distributions of F1-score for predicting failures in disk model A2. The top 141 teams uploaded their solutions 405 times in total. We notice that only 4.3% teams achieve the F1-scores higher than 30%, which is much worse than that for disk model A1. The reason may be mainly on the late opening of the training and testing data of disk model A2, so the teams only have 5 days for in-depth analysis. Most teams had to quickly transfer their knowledge, features, and even models learned from disk model A1 directly to disk model A2. It may lead to the severe overfitting problems.

In semi-finals, 76 teams submitted 3,299 valid solutions to predict disk failures from the mixed disk models A1 and A2. Figure 4 shows that more than 61.8% teams obtain the F1-scores higher than 30%, which is much better than the results in preliminaries (19.5% teams for disk model A1 and 4.3% teams for disk model A2). It also indicates that after passing preliminaries, participants

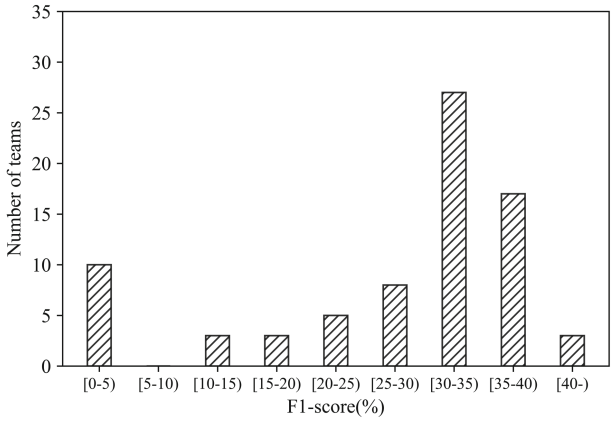


Fig. 4. F1-score distribution of prediction results of mixed disk A1 and A2 in semi-finals.

Table 3. Presentation results and semi-final scores of top teams.

Final ranking	Presentation score	Semi-final score
1	25.125	49.068
2	26.250	42.513
3	25.875	40.466
4	25.000	39.977
5	25.375	38.177
6	24.250	38.575
7	22.250	38.792
8	22.125	37.002
9	21.500	37.215
10	19.875	38.251
11	19.125	37.116

can pay more attention to feature engineering, modeling, and optimizing their solutions to improve their prediction results.

Finally, the top 12 teams entered the finals. Table 3 shows the presentation results and the semi-final score. The final score consists of 30% of presentation results and 70% of semi-final score. The presentation results in finals are evaluated in four major aspects, including novelty (10 points), reasonability (10 points), integrity (5 points), and presentation performance (5 points), by a committee of experts from academia and industry.

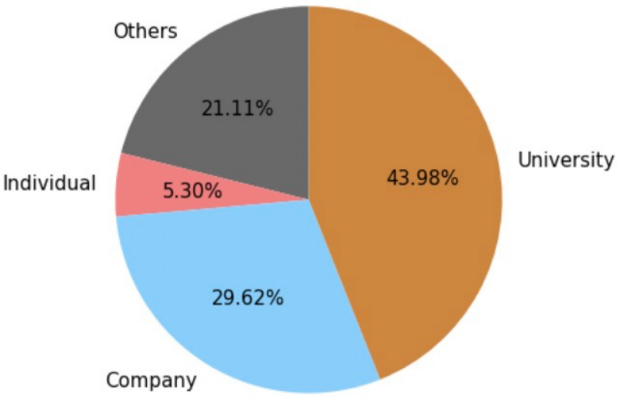


Fig. 5. Occupation analysis from registration information

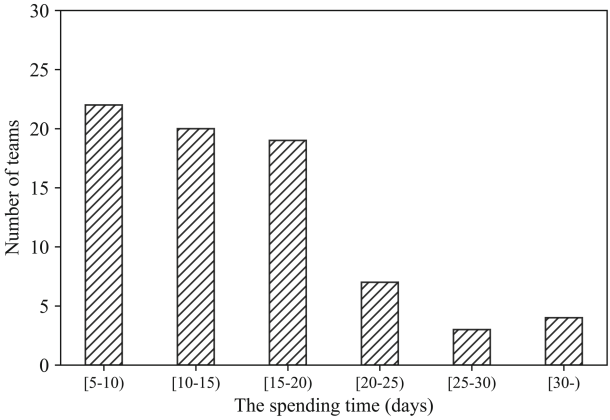


Fig. 6. Days spent to reach 85% of each team's best prediction results

4.2 Team Affiliation and Time Spent

We next present the team affiliation. Figure 5 shows that more than 70% of teams come from universities and companies. Most of them are familiar with machine learning and data mining, while very few of them have a strong background in storage reliability.

We also analyze the time spent of participants on the competition in order to know how much time participants need to apply their knowledge into a new field. Figure 6 shows the histogram results of days spent when reaching 85% of the best prediction accuracy for a team. We choose the 85%-mark as it can reflect that participants have completed the most part of their solutions. We notice that more than 81.3% of the teams can complete the majority of the task within 20 days, while around 29.3% of the teams spend less than 10 days on the task. These results provide us very useful insights for our future promotion of AI OPs to the community.

5 Mainstream Methodology and Highlighted Techniques

In this section, we summarize the techniques and methods applied in the competition based on the reviewed submissions.

5.1 Mainstream Methodology

From the reviewed submissions, the workflow of most solutions is comprised of four components, including data preprocessing, training sample generation, feature engineering, and modeling.

– **Data preprocessing.** As we describe before, there exists data noise in the SMART logs and failure labels. Thus, data preprocessing becomes an essential step that is applied by almost all teams. Most of the teams apply simple

methods to solve the problem. For example, they drop the samples with missing data directly or interpolate missing data by forward filling or backward filling.

- **Training sample generation.** Re-sampling techniques [3] are popular approaches for dealing with data imbalance. Existing methods include *oversampling* (e.g., SMOTE [7]), which directly duplicates positive samples, and *undersampling* (e.g., cluster-based undersampling [23]), which selects a subset of negative samples randomly with a predefined ratio.
- **Feature engineering.** Feature construction and feature selection are two important steps for feature engineering. In the competition, almost all teams exploit sliding-window-based statistical features with various window lengths, such as difference, mean, variance, and exponentially weighted moving-average values. Some teams also select important features based on correlation analysis and remove the weakly correlated SMART attributes to failures.
- **Modeling.** Most teams formulate disk failure prediction as a binary classification problem and use tree-based ensemble models, such as random forests and the decision-tree variations. Among them, LightGBM [14] and Xgboost [8] are applied the most because of their efficiency in execution time and memory usage.

5.2 Highlighted Techniques and Novel Ideas

In addition to conventional approaches, we notice that participants also propose and try many novel ways in the competition. We highlight some approaches and categorize them into the aforementioned four components.

In data preprocessing, a team proposes the cubic spline interpolation method to solve data missing problem, and their experiment results show that it can improve the benchmark result of F1-score by more than 3%.

In training sample generation, two teams apply different methods from the above resampling techniques, i.e., GAN [10] and self-paced ensemble model [18]. GAN augments positive samples, while self-paced ensemble model is an undersampling method for downsampling negative samples. From the experimental results, these two methods become useful complements to the re-sampling techniques for mitigating the data imbalance problem.

In feature engineering, some teams propose different feature construction methods based on data analysis. They analyze the distance of failure occurrences, distributions of disk lifetime, and data missing ratio. We find that each of the methods, as well as the combinations of the methods, can improve the overall prediction results.

In modeling, in addition to using binary classification models, several teams formulate the problem as a multi-label classification or a regression problem, which can result in a higher F1-score. Furthermore, a team designs a two-layer stacking model, in which the second layer uses different features from the first layer and aims to reduce the number of false positives. All the highlighted creative methods give us more inspirations and will be helpful for all of us in future exploration.

6 Discussion

From the competition perspective, we admit that some tricks are very useful for improving prediction results. However, from an online deployment standpoint, besides the balance of solution complexity and online performance, we pay more attention to the interpretation of the model and results. Thus, some features and methods applied in this competition are debatable for production deployment. We list some of them for open discussion:

- *Should we take SMART-9 (power-on hours) as one of the important features?* The raw value of SMART-9 is a cumulative value and indicates the lifetime of a disk. This value should increase by 24 h for normal disks if we collect the data on a daily basis. Also, we do not find any evidence in the production environment that the statistical features based on SMART-9 have a significant correlation with disk failures.
- *Should we do bitwise decoding for the SMART attributes, such as SMART-1 (read error rate), SMART-7 (seek error rate), SMART-188 (command timeout), and SMART-240 (head flying hours)?* The raw values of these SMART attributes are vendor-specific and are often meaningless as decimal numbers [4]. Some teams construct statistical features based on these SMART attributes without bitwise decoding, but we are still unsure whether this method is reasonable and effective.
- *How should we interpret tree-based ensemble models?* Tree-based ensemble models, like random forest and GBDT, are popular and widely used in the competitions. However, these models can only provide overall feature importances without clear information on the individual predicted output to support engineers for locating and solving disk failures.

Another interesting phenomenon is the limited usage of cutting-edge techniques like deep learning. One possible reason is that deep learning is a kind of data-hungry methodology. Even though we have opened the large-scale datasets, the data size is still insufficient for teams to build sophisticated deep learning neural networks.

Besides the techniques and methods mentioned and applied in this competition, we have published part of our progress in [11,12]. Although the results cannot be directly comparable with this competition because of the differences of datasets, techniques like data preprocessing (correlation analysis, spline interpolation, automated pre-failure backtracking, denoising etc.), feature engineering, and modeling are fully tested and applied in production environments.

In the future, we will keep pushing this area forward by gradually opening more anonymized datasets from different disk models, manufacturers, performance data, and system logs in addition to the SMART logs and failure tags. All datasets will be made available on the official Github website [1]. Also, other AI OPs tasks, such as memory error prediction, server downtime prediction, server cluster auto-healing, application-level intelligent operations, will also be taken into consideration. With more data and information, we encourage the

community to find more interesting and challenging problems in this field for further research and breakthrough.

Acknowledgement. We would like to express our gratitude to the PAKDD conference, especially Prof. Mengling Feng and Prof. Hady Wirawan Lauw for the great help and coordination. We thank Jingyi Huang, Ting Wang and Lele Sheng from the Tianchi platform for their hard work on test environment building, competition organization and coordination.

References

1. Alibaba Cloud AIOPs open datasets. https://github.com/alibaba-edu/dcbrain/tree/master/diskdata
2. Data Age 2025 - The Digitization of the World From Edge to Data. https://www.seagate.com/our-story/data-age-2025/
3. Resampling: Oversampling and Undersampling. https://en.wikipedia.org/wiki/Oversampling_and_undersampling_in_data_analysis
4. Wiki on S.M.A.R.T. https://en.wikipedia.org/wiki/S.M.A.R.T
5. Alagappan, R., Ganesan, A., Patel, Y., Arpaci-Dusseau, A.C., Arpaci-Dusseau, R.H.: Correlated crash vulnerabilities. In: Proceedings of USENIX OSDI (2016)
6. Botezatu, M.M., Giurgiu, I., Bogojeska, J., Wiesmann, D.: Predicting disk replacement towards reliable data centers. In: Proceedings of ACM SIGKDD (2016)
7. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**, 321–357 (2002)
8. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In Proceedings of ACM SIGKDD (2016)
9. Eckart, B., Chen, X., He, X., Scott, S.L.: Failure prediction models for proactive fault tolerance within storage systems. In: Proceedings of IEEE MASCOTS (2009)
10. Goodfellow, I., et al.: Generative adversarial nets. In: Proceedings of NIPS (2014)
11. Han, S., Lee, P.P.C., Shen, Z., He, C., Liu, Y., Huang, T.: Toward adaptive disk failure prediction via stream mining. In: Proceedings of IEEE ICDCS (2020)
12. Han, S.: Robust data preprocessing for machine-learning-based disk failure prediction in cloud production environments. arXiv preprint arXiv:1912.09722 (2019)
13. Japkowicz, N.: The class imbalance problem: significance and strategies. In: Proceedings of ICAI (2000)
14. Ke, G., et al.: LightGBM: a highly efficient gradient boosting decision tree. In: Proceedings of NIPS (2017)
15. Li, J., et al.: Hard drive failure prediction using classification and regression trees. In: Proceedings of IEEE/IFIP DSN (2014)
16. Li, J., Stones, R.J., Wang, G., Li, Z., Liu, X., Xiao, K.: Being accurate is not enough: new metrics for disk failure prediction. In: Proceedings of IEEE SRDS (2016)
17. Li, P., Li, J., Stones, R.J., Wang, G., Li, Z., Liu, X.: ProCode: a proactive erasure coding scheme for cloud storage systems. In: Proceedings of IEEE SRDS (2016)
18. Liu, Z., et al.: Self-paced ensemble for highly imbalanced massive data classification. arXiv preprint arXiv:1909.03500 (2019)
19. Lu, S.L., Luo, B., Patel, T., Yao, Y., Tiwari, D., Shi, W.: Making disk failure predictions SMARTer! In: Proceedings of USENIX FAST (2020)

20. Wang, G., Zhang, L., Xu, W.: What can we learn from four years of data center hardware failures? In: Proceedings of IEEE/IFIP DSN (2017)
21. Xiao, J., Xiong, Z. , Wu, S. , Yi, Y., Jin, H., Hu, K.: Disk failure prediction in data centers via online learning. In: Proceedings of ACM ICPP (2018)
22. Xu, Y., et al.: Improving service availability of cloud systems by predicting disk error. In: Proceedings of USENIX ATC (2018)
23. Yen, S.-J., Lee, Y.-S.: Cluster-based under-sampling approaches for imbalanced data distributions. Expert Syst. Appl. **36**(3), 5718–5727 (2009)
24. Zhu, B., Wang, G., Liu, X., Hu, D., Lin, S., Ma, J.: Proactive drive failure prediction for large scale storage systems. In: Proceedings of IEEE MSST (2013)