CSLE: A Cost-sensitive Learning Engine for Disk Failure Prediction in Large Data Centers

Xinyan Zhang[†], Kai Shan[†], Zhipeng Tan[*†], Dan Feng[*†]
[†]Wuhan National Laboratory for Optoelectronics, School of Computer Science and Technology
Huazhong University of Science and Technology, Wuhan, China
Email: {xyzhangcs, shankai1995, tanzhipeng, dfeng}@hust.edu.cn
Corresponding Email: (tanzhipeng, dfeng)@hust.edu.cn*

***Abstract*—As the principal failure in data centers, disk failure may pose the risk of data loss, increase the maintenance cost, and affect system availability. As a proactive fault tolerance technology, disk failure prediction can minimize the loss before a failure occurs. Whereas, a weak prediction model with a low *Failure Detection Rate* (FDR) and high *False Alarm Rate* (FAR) may substantially increase the system cost due to inadequate consideration or misperception of the misclassification cost. To address these challenges, we propose a cost-sensitive learning engine *CSLE* for disk failure prediction, which combines a two-phase feature selection based on Cohen's D and Genetic Algorithm, a meta-algorithm based on cost-sensitive learning, and an adaptive optimal classifier for heterogeneous and homogeneous disk series. Experimental results on real datasets show that the *AUC* of *CSLE* is increased by 2%-42% compared with the commonly used rank-sum test. *CSLE* can reduce the misclassification cost by 52%-96% compared with the rank model. Besides, *CSLE* has a better pervasiveness than the traditional prediction model, it can reduce both the misclassification cost and the *FAR* by 16%-70% for heterogeneous disk series, and increase the *FDR* by 3%-29% for homogeneous disk series.**

***Index Terms*—cost-sensitive learning, failure prediction, machine learning, system availability**

I. Introduction

The rapid development of cloud computing and the growth of user data have led to a dramatic increase in the size of data centers. Availability assurance is critical for data centers as they need to provide services on a 24/7 basis. Since the disks are considered as the most frequently replaced hardware components in data centers such as Microsoft and Azure [1]–[3], the impact of their failures has become a priority for the designers and operators of data centers.

Different from traditional passive fault tolerance technologies (replica, Erasure Codes, and Redundant Arrays of Independent Disks, etc.) that handle disk failures once they occur, proactive fault tolerance technologies (such as failure prediction) aim to minimize the loss and ensure system availability before disk failures. *S.M.A.R.T* (Self-Monitoring, Analysis and Reporting Technology) [4], a failure detection, monitoring, and maintenance technology, allows manufacturers to provide advanced notification of disk failures. All disk manufacturers adopt the threshold-based algorithms based on *S.M.A.R.T* to measure reliability and detect disk failures. However, these algorithms' detection ability is weak because they are generally threshold-based and they aim to avoid expensive false alarm cost [5].

Many effective methods for disk failure prediction based on *S.M.A.R.T* have been proposed, including statistical methods and techniques based on traditional machine learning (*ML*) and deep learning (*DL*) [1], [2], [6]–[8]. Almost all of these methods aim to achieve high *Failure Detection Rate* (FDR) and low *False Alarm Rate* (FAR). However, the basic assumption of these methods is that the misclassification cost of all misclassified samples is the same. In actual application scenarios, the cost of classifying a healthy disk as a failed one is different from that of classifying an ”about-to-be-failing” disk as a healthy one. The misclassification of a healthy disk as a failed one can incur extra data migration overhead, including unnecessary space occupation, waiting delay, and extra network bandwidth, etc. If an ”about-to-be-failing” disk is misclassified as a healthy one, data may be lost and services may be unavailable. These two situations differ greatly in cost as data migration cost in data centers are high, but the probability of data loss is low. The number of healthy disk samples in real data centers is much larger than that of failed disk samples. In other words, the disk dataset is extremely imbalanced, and the difference between positive and negative samples has different effects on classification accuracy, which is also a classic problem in classification tasks. Cost-sensitive learning method can effectively alleviate data imbalance in classification tasks and has been applied to various traditional machine learning techniques. The researchers [3] propose a cost-sensitive ranking model to address the imbalanced data. The failure probability of all samples is sorted in descending order (named *Rank Model*), and then the top α samples with the highest failure probability are selected for failure prediction. However, this method is a post-processing method based on threshold modification. The prediction model is not truly cost-sensitive and the cost is not the lowest, so there is room for further optimization. Besides, massive heterogeneous disk series exist in real data centers due to different purchase times and batches. Traditional methods tend to build prediction models separately for homogeneous series, which undoubtedly increases the cost. We believe that the construction of homogeneous or heterogeneous prediction models for different evaluation indicators can inspire the availability maintenance of data centers.

To address the challenges mentioned, this paper proposes *CSLE*, a cost-sensitive learning engine for disk failure prediction to reduce the misclassification cost while maintaining high

978-3-9819263-6-1/DATE22/©2022 EDAA

FDR. This is a pre-processing method, and the constructed prediction model is cost-sensitive. It treats a specific classification algorithm as a black box, so it does not need to modify a specific classification algorithm and can adapt to a variety of algorithms. In summary, our contributions are as follows:

- To improve the classification performance of the model and reduce the number of features, we design a two-phase feature selection algorithm based on *Cohen's D* and *Genetic algorithm* (GA). It can yield the *AUC* that is 2% to 42% higher than that of the rank-sum test.
- To reduce the misclassification cost of the model, we propose a meta-algorithm based on cost-sensitive learning that is different from the post-process method. It costs 52% to 96% less than that of the rank model.
- The adaptive optimal classifier for homogeneous and heterogeneous disk series enables better model selection for different target metrics. Compared with traditional methods, it can reduce the misclassification cost by 16%-70% and the *FAR* by 16%-70% for heterogeneous disk series, and increase the *FDR* by 3%-29% for homogeneous disk series.

II. Related Work and Motivation

A. Related Work

Data centers regularly store disk information in the form of snapshots. *Backblaze*'s data center stores disk status information daily and opens it quarterly [9]. It provides massive original data for powerful data analysis tools such as ML and data mining [10]. The following two methods are commonly used for disk failure prediction in recent years.

(1) Statistical methods: The traditional *threshold-based* method aims to improve *FDR* by sacrificing *FAR*, so the *FDR* is only 3% to 10%, which is too conservative [5]. The researchers utilize statistical methods to improve *FDR* while keeping *FAR* low. It has the advantage of lower training and computational overheads. The researchers [11] use Mahalanobis distance (MD) to achieve 68% *FDR* and 0 *FAR*. In [6], the researchers achieve 60% FDR and 0.5% FAR via the rank-sum test, and these hypothesis testing methods can also be used for feature selection [5]. In [12], the researchers construct *Hidden Markov Models* (HMM) or *Hidden Semi-Markov Models* (HSMM) to predict disk failures. However, these methods ignore the subsequent influence of misclassification cost on model prediction results.

(2) Machine Learning/Deep Learning: Many *ML* algorithms are used to predict disk failures, such as *Decision Tree* [13], *Bayesian* [14], *SVM* [15], and *Artificial Neural Network* [15]. Combining *NB* algorithm and *EM* algorithm, *NBEM* algorithm [14] is proposed. *RNN* (Recurrent Neural Network) [10] is used to build residual life prediction models and achieve reasonable and accurate health status assessment. *S.M.A.R.T* data can be treated as time-series data, and *LSTM* (Long Short Term Memory network) is suitable for processing such sequential data [16].

None of these related works takes into account the cost of misclassification, which affects the system availability, nor do they validate the effectiveness of a common cost-sensitive model based on heterogeneous data.

B. Motivation

There are still challenges that need to be addressed in the field of disk failure prediction.

High Dimensionality of *S.M.A.R.T.* Each disk has nearly 45 *S.M.A.R.T* features, each with *Raw* and *Normalized* values. Not all features are associated with disk failures, so feature selection is required for dimension reduction. Feature selection algorithms can be divided into three categories: filtering, encapsulation, and embedding. Some researchers [5] utilize non-parametric statistical tests (rank-sum and reverse arrangements) for feature selection. For the data suitable for parametric test conditions, non-parametric test will make rank-sum test lose some potential information, reducing the test efficiency. Therefore, we propose an optimal two-phase feature selection algorithm (Cohen's D and GA) which has better prediction performance than the traditional rank-sum test.

Ill-consideration or Misperception of the Misclassification Cost. Due to inadequate consideration or misperception of the misclassification cost, a weak prediction model with low *FDR* and high *FAR* may substantially increase the system cost. Cost-sensitive learning is a solution to this problem, and it can also reduce the model's misclassification cost. A cost-sensitive ranking model is established in [3], but it is only a post-processing method, without considering the misclassification cost during training. In contrast, our method takes into account the cost of misclassification during training. In [17], the matrix cost is considered to obtain better classifier performance during training. The meta-algorithm based on cost-sensitive learning that we proposed can further reduce the cost. In [3], *FastTree* algorithm similar to *GBDT* is adopted. The ranking model is not universal in other *ML* algorithms. Rich probability is what the ranking model needs, but the decision tree cannot provide it. Therefore, we propose a sample weighted meta-algorithm combined based on cost-sensitive learning. It can be easily adapted to other classification algorithms and can further reduce the cost compared with the method in [3].

Weak Performance of the Generalized Model. Most of the current failure prediction studies focus on improving *FDR* and maintaining acceptable *FAR*, usually modeling individually for a single disk series rather than building a generalized model for heterogeneous data composed of multiple disk series. *FDR* can reach up to 98% for homogeneous disk series [2]. In [14], the authors use *Decision Tree*, *Neural Networks*, and *Logistic Regression* to build a generalized model to directly predict heterogeneous data. The generalized model based on *Decision Tree* perform best, with *FDR* of 52.20%, but it is still far lower than the FDR level achieved by a recent work which builds the model separately for homogeneous disk series. In [3], disk failure prediction is achieved based on cost-sensitive learning, but the performance comparison of the two modeling methods is not described in detail. However, for homogeneous or heterogeneous disk series data, it is difficult to determine whether to build a generalized model or a separate model under different evaluation metrics. Therefore, we implement an adaptive optimal classifier for homogeneous and heterogeneous disk series, enabling the model to better select different target metrics.

III. DESIGN OF CSLE

In this section, we first provide an overview of our proposed prediction engine *CSLE* in section III-A, and then introduce the approach on feature selection and model construction in section III-B and section III-C.

A. Scheme Overview

To reduce the misclassification cost of the disk failure prediction model for imbalanced datasets, we propose *CSLE* for disk failure prediction in data centers. Figure 1 presents the overview of *CSLE*.

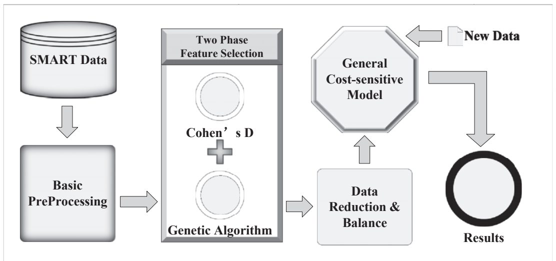

Fig. 1: The architecture of *CSLE*.

Basic preprocessing includes missing value processing and useless information filtering. The two-phase feature selection is described in subsection III-B.

Exponentially Weighted Moving Average (*EWMA*) method is adopted for data reduction. We split continuous data into several segments and then perform *EWMA* on each segment. Each segment is evaluated as an average value. The weight of data is positively correlated with the timeline. We use *K-means* to balance positive and negative samples. The label of the failed disk is set to positive. Specifically, for each failed disk, we replace its failure status with 1 instead of the original 0, which is called backtracking data for the previous n days. In this paper, we set n=5. *K-means clustering* is performed for data of healthy disks. For each cluster, the data sample closest to the center is selected as the representative of the cluster. When *k* is equal to the minority samples, a training set with balanced positive and negative samples can be generated.

The prediction model based on cost-sensitive learning method is listed in section III-C.

B. Two-Phase Feature Selection

We propose a two-phase algorithm list in algorithm 1 based on Cohen's D and *GA* for feature selection. The purpose of the rank-sum test is to find out the features that have an obvious difference in distribution between a failed disk and a healthy one. These features are considered to be capable of distinguishing a healthy disks from a failed one. To more accurately discover features with significant differences in positive and negative samples, Cohen's D is adopted for feature engineering. Cohen's D is an indicator reflecting the mean deviation [18]. Since data with different distributions also have mean deviation, Cohen's D can be used to select features with obvious distribution differences. Meanwhile, feature selection is a combinatorial optimization problem of *NP-hard*. The time complexity of the exhaustive method is $O(2^n)$, which is unacceptable for data with high dimensional features. Therefore, we use *GA* to solve this problem. As *GA* is a group-based meta-heuristic optimization algorithm, it has adaptive, self-searching, self-organizing and implicit parallelism characteristics [19]. It is very suitable for solving combinatorial optimization problems such as feature selection.

ALGORITHM 1: Two-phase Feature Selection

```
Input: the dataset $D$;
Initial population size $n\_population$;
The length of chromosome $chrom\_len$;
Iteration number $n\_iter$;
The num of chromosomes chosed in the Roulette $m$;
The variation ratio $\alpha$
Output: $chrom\_sets$ with the highest AUC value
while TFS($D, n\_population, chrom\_len, n\_iter, m, \alpha$)
 do
    $first\_result$ = [];
    for column_i in $D$.columns do
        $CD_i = q_a\sqrt{\frac{k(k+1)}{6N}}$;
        if $CD_i < CD_{threashold}$ then
            add $D.columns_i$ to $first\_result$;
        end
    end
    $chrom\_set$ = matrix_[$n\_population$][$chrom\_len$]
      whose elements are 0 or 1;
    for j in [0,n_iter] do
        for i in [0,n_population] do
            models[i] = the constructed ML model;
            aucs[i] = the AUC value of models[i];
        end
        parts1 = select $m$ chromosomes via $Roulette$;
        parts2 = chromosomes in the $chrom\_set$
          (randomly cross to generate ($n\_population$ -
          m) chromosomes);
        Choose $\alpha$ chromosomes from the union of
          parts1 and parts2 as variation;
    end
end
return $chrom\_set$ who has the highest AUC value
```

The two-phase feature selection algorithm firstly calculates the Cohen's D value of each feature, and then selects the feature set less than the threshold. Then the feature set filtered by Cohen's D is iterated continuously through genetic algorithm, and the *chromosome* with the highest AUC value is finally selected. The value of *chromosome* represents the final feature subset.

C. Cost-sensitive Learning

1) Cost-Sensitive Learning Meta-Algorithm: Cost sensitive learning algorithms are mainly divided into three categories. The first one is to directly construct a cost sensitive model and fits the cost sensitive function into the classifier [20]; The second one is to combine the cost minimization technique with the model trained by the ensemble method to form a cost sensitive model [21]. The third one is to directly apply the misclassification cost to the data set in the form of weight. It makes the classifier reduce the misclassification cost by changing the weight of data [22]. This paper adopts the third method of data weighting.

The misclassification cost is different in real scenarios. Therefore, it should be set according to the actual situation in failure prediction. We refer to the setup of the reference of [3]. The ratio of the cost of misclassifying a healthy disk as a failed one and the cost of misclassifying a failed disk as a normal one is set to 3:1 (*Cost_FP: Cost_FN*). The specific settings of *cost matrix* are shown in Table I. Entered as a parameter, the cost matrix is the key input of cost-sensitive algorithms.

TABLE I: Cost Matrix

		Prediction	
		P/Failure	N/Healthy
True Result	T/Failure	0	**1**
	N/Healthy	**3**	0

The algorithm 2 gives the details of the meta-algorithm based on cost-sensitive learning that we adopt. Its essence is to set weight for each sample instance. It treats the underlying classifiers as black boxes (with no need to understand the functionality of the basic classifiers or change to them).

ALGORITHM 2: Cost-Sensitive Learning Algorithm

```
Input: the dataset $D$,cost matrix $C$
Output: weight_instances
while CSLA($D, C$) do
    $weight_p$ = weight of positive instances;
    $weight_n$ = weight of negative instances;
    weightOfInstancesInClass = [$weight_p$,$weight_n$];
    $\sum_1^n Weights[i]$ = the sum of weights of all
      instances;
    $c_{num}$ = the num of class (set it as 2);
    for i in [0,$c_{num}$] do
        for j in [0,$c_{num}$] do
            $\sum_1^n MissClassWeights[i]$ += C[i,j];
        end
        $\sum_1^n WeightFactor[i]$ +=
          $\sum_1^n MissClassWeights[i] \times$
          $\sum_1^n Weights[i]$;
        $WeightFactor[i] = \frac{\sum_1^n MissClassWeights[i] \times \sum_1^n Weights[i]}{\sum_1^n WeightFactor[i]}$
    end
    for each instance in instances do
        $weight\_instances[i] = weightFactor[i] \times$
          $instance.weight$;
    end
end
return weight instances
```

2) Algorithm Comparison: Many *ML* schemes use different datasets, features, and data preprocessing techniques. But for some reason, they do not make their own datasets public. Therefore, it is difficult to objectively evaluate the effectiveness of *ML* algorithms in hard disk failure prediction. Although [16], [17] have compared some algorithms, the former only compares 21 algorithms based on a dataset with a small volume and the two algorithms only compare on the test set. Whereas, the expected result is the generalization performance of each model, which is different from the performance on the test set. The selection of test sets and the randomness of some machine learning algorithms make performance comparisons based on the test sets tend to be unstable.

Therefore, we adopt *Friedman Test* [23] to compare the performance of 20 algorithms representing different *ML* algorithms. The algorithms we compare can be divided into 8 categories. (1) Decision Tree Classification: ExtraTreeClassifier, DecisionTree (C4.5 and CART), (2) Neighbor Classification: NearestCentroid, KNN, (3) Linear Model: PassiveAggressiveClassifier, SGDClassifier, Perceptron, Logistic Regression (LR), (4) Bayes Classification: BernoulliNB, MultinomialNB, NaiveBayes (NB), (5) Ensemble Classification: GBDT, AdaBoost, Xgboost, LightGBM, RandomForest, (6) Discriminant Analysis: QuadraticDiscriminantAnalysis, (7) Neural Network: Multi-layer Perceptron (MLP), (8) SVM: SVM. A simple *ZeroR* algorithm is also considered as a reference. The predicted result is the label with the highest probability of occurrence. These 20 algorithms almost represent the commonly used *Ml* classification algorithms. Due to the relatively poor interpretability of the advanced neural networks of time series such as RNN and LSTM, these characteristics are inconsistent with our original intention to pursue high efficiency, simplicity, and good interpretability, so we do not compare RNN and other advanced neural networks.

The test results show that *Random Forest* has good prediction performance, which is consistent with the conclusion of [17]. We take the *Random Forest* as a baseline to compare with our solutions.

IV. Evaluation

In this section, we evaluate our scheme *CSLE* in terms of feature selection, model comparison, and the best choice for homogeneous or heterogeneous datasets in large data centers.

A. Setup

TABLE II: The Distribution of dataset

Source	Dataset	Time	No.Healthy	No.Failed	No.Instances
Backblaze	S4D	2017	34128	1061	12237700
		2018_Q1	30763	178	2853568
		2018_Q2	27342	134	2675525
	S8D	2017	9886	93	3523493
		2018_Q1	9870	21	888774
		2018_Q2	9861	25	899939
	S8N	2017	14421	88	14509
		2018_Q1	14372	28	1293557
		2018_Q2	14369	24	1309622
	HGS	2017	16162	89	5154696
		2018_Q1	15323	16	1363173
		2018_Q2	15051	10	1384015
Murray Set			178	191	68411

Dataset. We mainly use open source data from *Backblaze* data center. For some disk series, we use annual 2017 data for training, Q1 2018 data for validation, and Q2 2018 data for testing. The distribution of *Backblaze* data we use is shown

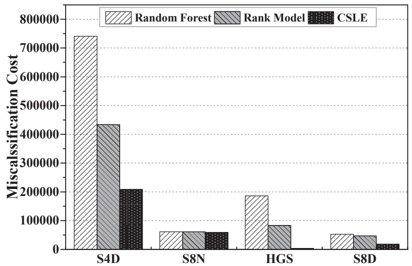

Fig. 2: Lower *Cost* of CSLE

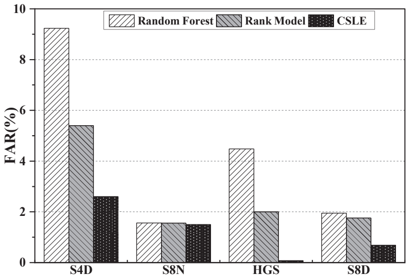

Fig. 3: Lower *FAR* of CSLE

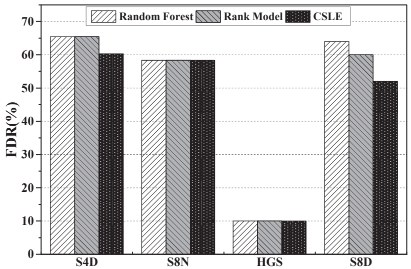

Fig. 4: Relatively stable *FDR* of CSLE

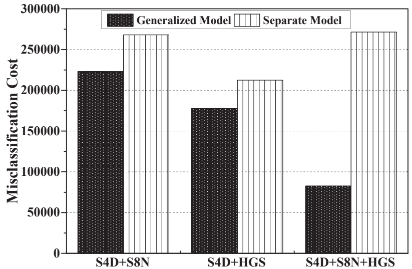

Fig. 5: Lower *Cost* of generalized model

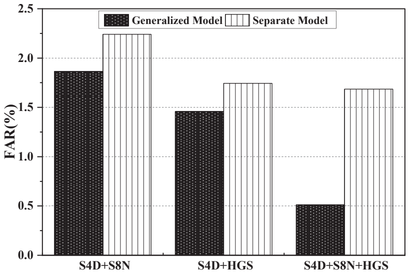

Fig. 6: Lower *FAR* of generalized model

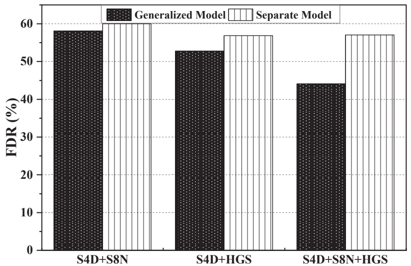

Fig. 7: Higher *FDR* of separate model

in table II. For simplicity, we use S4D, S8D, S8N and HGS to represent the dataset of ST4000DM000, ST8000DM002, ST8000NM0055, and HGST HMS5C4040BLE640 series. We also utilize *Murray Set* used in [17].

Evaluation Metric. The confusion matrix can fully characterize the performance of the classifiers. *TP* (True Positives) and *TN* (True Negatives) indicate the number of instances with correct classification. *FP* (False Positives) and *FN* (False Negatives) represent the number of misclassified instances, indicating different error types.

FDR (=TP/(TP+FN)) and *FAR* (=FP/(TN+FP)) are two main metrics to evaluate the quality of the prediction model. *AUC* (Area Under the ROC Curve) is often used as a trade-off between *FDR* and *FAR*. Since misclassification cost is clearly defined in our work, we can compare the performance of models according to the cost, which is equivalent to finding the lowest cost point on *ROC* curve.

Comparative Object. Random Forest algorithm is used as the test baseline in this paper. *CSLE* is compared with the rank model which is cost-sensitive for disk failure prediction to verify whether the cost of the proposed model is lower.

B. Feature Selection

We conduct a comparative experiment, in which the other processing steps adopt the same method except for different feature selection algorithms. To highlight the role of feature selection, we select three commonly used classifiers for disk failure prediction according to comparison results in section III-C2.

The bold data in table III show that our feature selection algorithm is superior to the rank-sum test for the metric of *AUC*. The *AUC* of the proposed two-phase feature selection increases 2%-42%. Due to the randomness of *GA*, the performance of the second step (GA) is not always guaranteed to be better than that of the first step (*Cohen's D*) . But our first step is almost identical to the rank-sum test. When the genetic algorithm does not work well, we abandon *GA* and just take the first step, which has the same effect as the rank-sum test.

TABLE III: The comparison of feature selection

Dataset	method	LR	MLP	NB
Murray Set	rank-sum test	0.8462	0.4664	0.6275
	Cohen's d	0.8492	0.5302	0.6292
	Cohen's d+GA	**0.862**	**0.8814**	**0.9738**
S4D	rank-sum test	0.5903	0.5	0.5758
	Cohen's d	0.6001	0.5102	0.5756
	Cohen's d+GA	**0.611**	**0.7656**	**0.7625**
S8D	rank-sum test	0.52	0.5	0.4995
	Cohen's d	0.5238	0.5002	0.5
	Cohen's d+GA	**0.8125**	**0.8251**	**0.8077**

C. Model Comparison

In Figure 2, the cost of *CSLE* is 52%-96% lower than that of the rank model used in [3]. The setting of the cost matrix (*Cost_FP: Cost_FN=3:1*) makes *CSLE* achieve lower *misclassification cost*. In Figure 3, the *FAR* of *CSLE* is much lower than that of the rank model. The setting of the cost matrix we used is to encourage *CSLE* to keep *FAR* low. Figure 2 and Figure 3 have the same trend, because low cost is achieved by low *FAR*. The lower *FAR* of the model is, the lower *FDR* is. In Figure 4, except for the ST8000DM002 model, *CSLE* keeps *FDR* unchanged or slightly lower.

D. Generalized or Separate Model?

Many previous works, such as [2], [6], have built models for homogeneous disk series separately in the hope of achieving high *FDR*. In *CSLE*, the adaptive optimal classifier for homogeneous and heterogeneous disk series enables better model selection for different target metrics. We compare different evaluation metrics (*FDR, FAR*, and *misclassification cost*) for these two strategies.

As shown in Figure 5 and Figure 6, it is better to build a generalized model for homogeneous disk series if we focus on minimizing the misclassification cost or the *FAR* (both reduced by 16%-70%), otherwise building separate model for homogeneous disk series if we focus on the *FDR* (increased by 3%-29%), which is shown in Figure 7.

E. Time In Advance

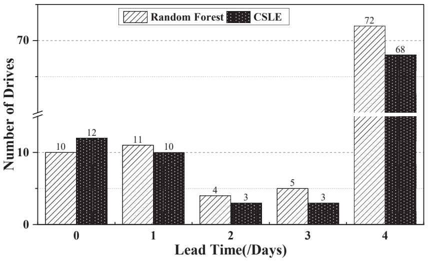

Fig. 8: The Lead Time Comparison.

Apart from metrics such as *FAR*, *FDR*, and the misclassification cost, we also focus on the lead time before failures (called *Lead Time*) to provide sufficient time for data protection measures. Figure 8 presents the *Lead Time* distribution of *CSLE*. For the test set of S4D, the *Random Forest* model can predict 68% of the failed disks at least one day in advance, and detect 76% of the failed disks in total. And *CSLE* we proposed can predict 62% of the failed disks at least one day in advance, and detect 71% of the failed disks in total. Because the cost matrix in *CSLE* is designed to make the model have a lower *FAR*, the *CSLE* model has a lower *FDR* than the *Random Forest* model. In the target backtracking step, we set n = 5, so the lead time does not exceed 5.

V. Conclusion

Misclassification of the disk failure prediction model may increase unnecessary migration overhead, pose the risk of data loss, and affect system availability. In this paper, we propose *CSLE* to minimize the misclassification cost and reduce the *FAR*. We design a two-phase (*Cohen's D + GA*) feature selection algorithm, develop a cost-sensitive learning-based model, and provide an adaptive optimal classifier for homogeneous and heterogeneous disk series. Experimental results on real datasets show that *CSLE* can increase the *AUC* by 2%-42% and reduce the misclassification cost by 52%-96%. *CSLE* provides better model selection for different target metrics. It can reduce the misclassification costs and the *FAR* by 16%-70% by constructing a generalized model for homogeneous disk series, and increase the *FDR* by 3%-29% by building a separate model for homogeneous disk series.

Acknowledgment

This work was supported by the National Natural Science Foundation of China No.61821003, No.61832007, No.61772222, No.U1705261, Major Project of National Science and Technology of China No.2017ZX01032-101, and the Fundamental Research Funds for the Central Universities No.2019kfyXMBZ037. This work was also jointly completed with Key Laboratory of Information Storage System and Engineering Research Center for Data Storage Systems and Technology, Ministry of Education, China.

References

[1] B. Schroeder and G. A. Gibson, "Disk failures in the real world: What does an mttf of 1, 000, 000 hours mean to you?" in *FAST*, vol. 7, no. 1, 2007, pp. 1–16.

[2] M. M. Botezatu, I. Giurgiu, J. Bogojeska, and D. Wiesmann, "Predicting disk replacement towards reliable data centers," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 39–48.

[3] 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 *2018 {USENIX} Annual Technical Conference ({USENIX}{ATC} 18)*, 2018, pp. 481–494.

[4] B. Allen, "Monitoring hard disks with smart," *Linux Journal*, no. 117, pp. 74–77, 2004.

[5] J. F. Murray, G. F. Hughes, K. Kreutz-Delgado, and D. Schuurmans, "Machine learning methods for predicting failures in hard drives: A multiple-instance application." *Journal of Machine Learning Research*, vol. 6, no. 5, 2005.

[6] G. F. Hughes, J. F. Murray, K. Kreutz-Delgado, and C. Elkan, "Improved disk-drive failure warnings," *IEEE transactions on reliability*, vol. 51, no. 3, pp. 350–357, 2002.

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

[8] J. Zhang, K. Zhou, P. Huang, X. He, Z. Xiao, B. Cheng, Y. Ji, and Y. Wang, "Transfer learning based failure prediction for minority disks in large data centers of heterogeneous disk systems," in *Proceedings of the 48th International Conference on Parallel Processing*, 2019, pp. 1–10.

[9] Backblaze, "Hard drive data and stats," 2017. [Online]. Available: https://www.backblaze.com/b2/hard-drive-test-data.html

[10] C. Xu, G. Wang, X. Liu, D. Guo, and T.-Y. Liu, "Health status assessment and failure prediction for hard drives with recurrent neural networks," *IEEE Transactions on Computers*, vol. 65, no. 11, pp. 3502–3508, 2016.

[11] Y. Wang, Q. Miao, E. W. Ma, K.-L. Tsui, and M. G. Pecht, "Online anomaly detection for hard disk drives based on mahalanobis distance," *IEEE Transactions on Reliability*, vol. 62, no. 1, pp. 136–145, 2013.

[12] Y. Zhao, X. Liu, S. Gan, and W. Zheng, "Predicting disk failures with hmm-and hsmm-based approaches," in *Industrial Conference on Data Mining*. Springer, 2010, pp. 390–404.

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

[14] G. Hamerly, C. Elkan *et al.*, "Bayesian approaches to failure prediction for disk drives," in *ICML*, vol. 1, 2001, pp. 202–209.

[15] B. Zhu, G. Wang, X. Liu, D. Hu, S. Lin, and J. Ma, "Proactive drive failure prediction for large scale storage systems," in *2013 IEEE 29th Symposium on Mass Storage Systems and Technologies*, ser. MSST '13, May 2013, pp. 1–5.

[16] J. Zhang, P. Huang, K. Zhou, M. Xie, and S. Schelter, "Hddse: Enabling high-dimensional disk state embedding for generic failure detection system of heterogeneous disks in large data centers," in *2020 {USENIX} Annual Technical Conference ({USENIX}{ATC} 20)*, 2020, pp. 111–126.

[17] T. Pitakrat, A. Van Hoorn, and L. Grunske, "A comparison of machine learning algorithms for proactive hard disk drive failure detection," in *Proceedings of the 4th international ACM Sigsoft symposium on Architecting critical systems*, 2013, pp. 1–10.

[18] C. J. Ferguson, "An effect size primer: a guide for clinicians and researchers." 2016.

[19] S. Sivanandam and S. Deepa, "Genetic algorithms," in *Introduction to genetic algorithms*. Springer, 2008, pp. 15–37.

[20] U. Knoll, G. Nakhaeizadeh, and B. Tausend, "Cost-sensitive pruning of decision trees," in *European Conference on Machine Learning*. Springer, 1994, pp. 383–386.

[21] P. Domingos, "Metacost: A general method for making classifiers cost-sensitive," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, 1999, pp. 155–164.

[22] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan, "Adacost: misclassification cost-sensitive boosting," in *Icml*, vol. 99. Citeseer, 1999, pp. 97–105.

[23] D. W. Zimmerman and B. D. Zumbo, "Relative power of the wilcoxon test, the friedman test, and repeated-measures anova on ranks," *The Journal of Experimental Education*, vol. 62, no. 1, pp. 75–86, 1993.