Записки научных
семинаров ПОМИ
Том 540, 2024 г.

K. Lukyanov, M. Drobyshevskiy, D. Turdakov

DETECTING AND ELIMINATING COVARIATE SHIFTS IN DATA FOR A MORE ROBUST HDD FAILURE PREDICTION

Abstract. Prediction of HDD failures has garnered significant attention in research, yet the persistence of covariate shifts in data remains a practical challenge. In this work we introduce a novel approach to training covariate shift detection models without the need for additional real data or artificial shift modeling. Moreover, we propose a comprehensive methodology integrating shift detection, administrator alerts, shift elimination, and HDD failure prediction. Experimental results demonstrate the viability of our real-world implementation.

§1. Introduction

Rapid growth of computational power and the vast amount of available information have transformed machine learning into an indispensable tool for various applications across different fields [1]. In recent years, machine learning has been increasingly utilized for addressing various practical tasks, including modeling issues in manufacturing, anomaly detection, equipment failure prediction, and more [3–6].

To address failure prediction tasks and anomaly detection in production, one can utilize time series analysis or classification on data obtained by averaging data over a specific time window, depending on the task's specifics [7–9].

However, despite achieving high quality in laboratory conditions, many prediction models may show significantly worse results in practice. One reason for this could be the presence of a covariate shift in the data due to various external factors, both intentional and natural [10].

Covariate shift occurs when the distributions of feature values in the training and test sets have different parameters (such as mean, variance, and others) [2]. Covariance in this context refers to the feature values.

Key words and phrases: HDD failure prediction, detecting and eliminating covariate shift.

Therefore, it is important to have built-in mechanisms for detecting shifts, notifying production operators, and attempting automatic calibration to mitigate the shift. In this work we examine the industrial application of machine learning models for predicting hard drive failures, investigate the impact of covariate shifts on prediction models, and explore trusted models capable of detecting and mitigating data shifts when possible.

When training a shift detection model, access to data with the shift is necessary, but obtaining it in practice can be challenging. Considering all real scenarios in modeling is difficult, and obtaining real data requires significant time and resources.

Our main contributions are the following:

- we present an algorithm that encompasses three crucial stages: detecting shifts, eliminating them, and effectively resolving prediction problems;
- we propose a novel approach to train a shift detection model without the need for additional shifted data;
- we conduct a comprehensive study on the impact of shifts and eliminating data on the quality of prediction models.

In the rest of the paper, we first review existing literature concerning the challenge of predicting HDD failures, as well as the detection and rectification of covariate shifts in data (see Section 2). Section 3 considers the dataset, models, and algorithms used in this work. Our experiments are discussed in detail in Section 4. Finally, Section 5 concludes the paper.

§2. Related work

Many algorithms have been developed to detect anomalies and predict failures. Their applications range from bearing fault detection [11, 12] to financial crisis prediction [13–15]. Most of these applications focus on short-term failure prediction. There is also another failure prediction group that focuses on long-term prediction [16–19]. However, the accuracy of long-term forecasting largely depends on the failure physics model (or the so-called degradation model). In our application, hard drive failure can be caused by many mechanisms, making it difficult to model the physics of failure. On the other hand, the purpose of our HDD failure prediction is to provide a short-term (i.e., 24–48 hours [20]) prediction that can provide users with sufficient margin to back up their data, so long-term prediction is not necessary. Detailed reviews of prediction methods and anomaly detection can be found in [21–24].

The problem of predicting HDD failure has a long history of exploration. Initially, early approaches heavily relied on various statistical methods [25–27]. With the advent of machine learning's popularity, it emerged as a prominent solution for HDD failure prediction [28–31]. Presently, the literature encompasses solutions using fundamental machine learning models, alongside with algorithms employing neural networks. Moreover, more intricate models featuring multiple stages of operation have also been proposed, which can combine some approaches [32, 33].

However, as mentioned in the previous section, the practical performance of models may decrease due to the presence of a covariate shift in the data. Covariate shifts can have different origins and require various mitigation methods. In the context of predicting HDD failure, common approaches in the literature include various statistical methods, distance estimation techniques, additional classifiers, as well as ensembles formed by combining multiple methods [34–36].

§3. Dataset preparation, HDD failure prediction, covariate shift modeling, detection, and eliminating

In this section, we present the findings of our research on predicting HDD failures and the challenges related to covariate shifts specific to artificial intelligence technologies applied to HDD datasets.

3.1. Dataset. The dataset comprises information about hard drives, encompassing indicators such as the date, disk serial number, model, memory size, and labels denoting correct operation. Additionally, it contains 254 SMART indicators, each with normalized values within ranges of 0–100, 0–200, or 0–253, contingent on the disk model. The data spans from January 1, 2020 to December 31, 2020, with the daily count of disks ranging from 124,000 to 166,000, and 1,498 disk failures recorded throughout the year.

In literature, researchers commonly used 5 to 20 key indicators or auxiliary models to select features. As this study primarily focuses on investigating the impact of covariate shift and integrating detection, shift elimination, and HDD failure prediction, only 7 SMART indicators out of 255 (5, 9, 187, 188, 194, 197, and 198) were utilized. These indicators include metrics such as the reallocated sector count, power-on hours, reported uncorrectable errors, command timeouts, temperature (in Celsius), current pending sector count, and uncorrectable sector count. Section 4 demonstrates that reducing the number of features did not significantly affect the quality of prediction models compared to those described in the literature.

The work considers only HDD with all indicator values in the range from 0 to 100. The focus primarily lay on the 0-100 range due to its larger representation, ensuring statistical significance in the experiments. To enable machine learning analysis, a balanced subsample was created by randomly selecting five records of normally operating disks each day.

3.2. Quality metric. For the classification experiments, we utilized the F1 metric implemented in the sklearn library. This metric was chosen because it considers both precision and recall simultaneously, providing a more comprehensive evaluation of model quality, particularly in experiments involving covariate shifts. Additionally, when addressing the HDD failure prediction problem, we used the accuracy metric to compare results with the baseline models described in the literature.

3.3. Generalized algorithm for predicting HDD failure. The general algorithm for predicting HDD failure operates as follows (see Fig. 1):

(1) data preparation: fill in missing values in the data and sort;
(2) check the normalization range;
(3) if the range is normal, then the Random Forest (RF) algorithm checks the data for covariate shift;
(4) if the range check fails, issue a strong warning to the server administrator. If RF detects a shift, issue a weak warning;
(5) if the data was transmitted in a group, launch statistical methods to eliminate the shift;
(6) after eliminating the shift or in its absence, launch the method for predicting HDD failure;
(7) provide prediction, raw data, warnings, and adjusted data, if available.

3.4. Basic HDD failure prediction. We consider the problem of predicting HDD failure as a binary classification task, utilizing three popular methods: linear regression [37], gradient boosting [38], and neural networks [39]. These methods were chosen due to their widespread use and

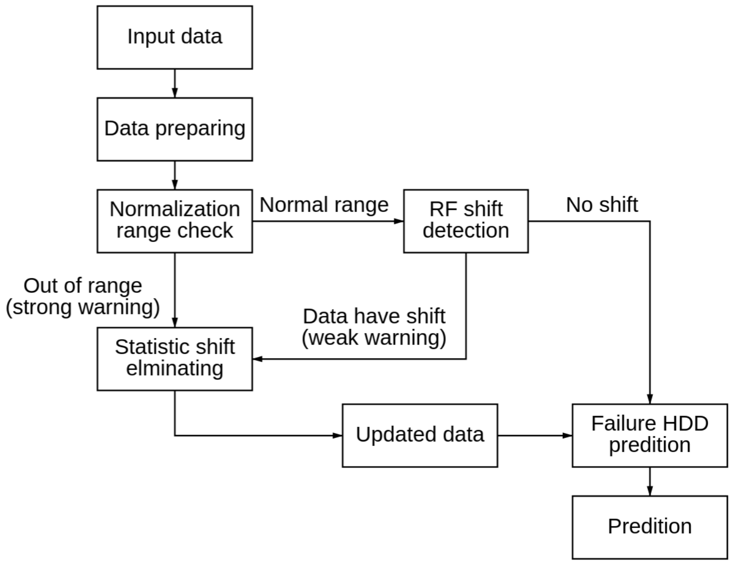

Figure 1. Generalized algorithm for predicting HDD failure.

effectiveness. Given that the focus of this work on studying and addressing covariate shift, we consider these well-established methods to be suitable and sufficient. However, in a generalized algorithm in the future, simpler methods can be replaced by more advanced ones to obtain better prediction performance.

The problem can be framed as anomaly detection in a time series or detecting shifts in trends. However, given that failed disks are promptly replaced with functional ones and are subsequently absent from the datasets, and that key SMART indicators do not deviate significantly from the norm (even on the day of disk failure), long-term prediction proves to be exceedingly challenging. Therefore, treating the problem as a time series analysis to pinpoint a change in trend is inappropriate. The significant degradation of indicators shortly before failure suggests that it is more advantageous to aggregate the time series indicators over a narrow time window (1–3 days) and then address the classification problem using these averaged indicators.

After selecting the hyperparameters on validation, the accuracy metric values were calculated as follows: 81.29%±0.01%, 83.27%±0.01%, and 82.11%±6.2% for the three models respectively. These accuracy rates are comparable to those in the literature.

More details about the configuration of hyperparameters of prediction algorithms are given in the next section (Section 4).

3.5. Covariate shift modeling. Covariate shifts can be divided into six groups: a change in the statistical properties of individual features, a shift in the feature space, a change in the domain, non-stationarity of features as a result of changes in the statistical properties of features over time, a systematic error in sample selection, and the batch effect as a result of introducing a systematic error at the stages of data collection or preliminary processing. In this work, we focused on modeling changes in the statistical properties of individual features and shifts in the feature space.

To evaluate the impact of the shift, perturbations were applied to the test sample of the model. Specifically, values equal to 5%, 10%, and 30% of the average characteristic value were added or subtracted from all points in the test sample. The number of features that shifted were 1, 3, and 7, respectively. When one or three features are being shifted, it corresponds to alterations in individual features, while all seven features being shifted corresponds to a change in the entire feature space.

3.6. Covariate shift detection. To detect a covariate shift, various methods have been employed in literature, including statistical tests, data visualization, training an additional model to evaluate the presence of a shift, or combinations of these approaches. However, visualization cannot be suitable for our study since the process must be automated. Additionally, since it is important to predict HDD failures for individual samples or small data groups, traditional statistical methods may not be adequate. Although limiting group sizes could be considered, it may restrict the algorithm's applicability.

Therefore, in this study we adopt an approach that involves training an additional model. Given that all disks have strictly fixed normalization boundaries, we also constrain the feature space within a compact range corresponding to the normalization range, specifically from 0 to 100 for all 7 features.

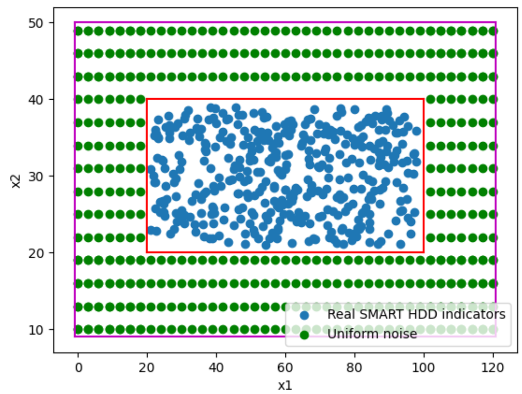

Figure 2. Principle of data generation for training a covariate shift detection model.

Typically, a detection model is trained to identify shifts. In practice, if such a model can effectively distinguish between new data and original data, it indicates a shift in the data; otherwise, the algorithm concludes that no shift has occurred. However, obtaining data with shift presents challenges. The original dataset is not shifted by default, so it is necessary to model the shifts. Yet, this approach may not accurately reflect real-world changes. Alternatively, real data already subjected to alterations can be used. However, gathering such additional data demands extra resources and extends the deployment time for the model.

We propose using an additional regular grid for training. From the dataset, we determine the minimum and maximum boundaries. The compact where the data is located remains unchanged, with no additional points generated within it. Next, for each feature, the boundary is expanded by 25% relative to the range between the maximum and minimum values of the feature, and a regular grid is generated within this expanded area. The step between points is chosen so that the number of generated points matches that inside the compact for each feature. If the minimum and maximum values result in fewer than 4 steps between grid points, the boundary is expanded to ensure at least one point relative to the minimum and maximum boundaries for each feature. In Fig. 2, real SMART indicators from the dataset are represented in blue, while the boundary of the real data compact is shown in red. The generated regular grid outside the compact is highlighted in green. The margin from the boundary of the compact is set at 25% of the difference between the boundaries of the compact for each coordinate, respectively. Although only two features are displayed in the figure for simplicity, a similar approach is applied to all seven features.

Consequently, a new dataset is generated alongside a regular grid outside the original data compact. Subsequently, a classification model, specifically the random forest model, is trained on this dataset.

In summary, the data is first checked to ensure that the normalization range is correct; if it does not fall within the specified range, it indicates a shift. Moreover, this implies that the prediction model received data from disks with a different normalization range, and the current HDD failure prediction model was trained on other data, rendering its prediction incorrect. If the output data has a similar normalization, it is fed to the random forest model, which is also responsible for detecting covariate shifts. If the model output indicates no shift, then the data is forwarded to the HDD failure prediction model.

3.7. Eliminating covariate shift. Standard approaches to addressing covariate bias include statistical methods, data weighting, domain adaptation, and dynamic model updating. However, weighing is not suitable for the task at hand, as individual predictions for each disk are crucial. Similarly, domain adaptation does not align with the shifts under consideration.

This article will explore the statistical method by considering all data in an amount equal to the average between the data used to train the prediction models and the biased test data. Removing a shift for individual data points is extremely challenging. In such cases, it becomes imperative to focus more on shift detection and promptly notify the server administrator.

§4. Experiments

This section presents the results of experiments with models for predicting HDD failure, the influence of shift, the detection of shift, and its elimination.

We have made the code available on GitLab[1]. It can be used to conduct experiments with other data that use CSV format.

4.1. Selection of optimal hyperparameters for an HDD performance prediction model. In the experiments aimed at studying models for classifying disks into normal and faulty categories, each group of disk records was divided into training, validation, and testing sets in the ratio of 80%, 5%, and 15%, respectively. When conducting experiments involving parameter variations, each experiment was repeated 25 times. In cases where determining the optimal hyperparameters was not straightforward, the experiment was performed 50 times. If ambiguity persisted after these repetitions, the hyperparameter value resulting in the best metric performance was selected, without considering the variance of the resulting model.

4.1.1. *Selection of optimal hyperparameters for logistic regression.* The optimal configuration, determined by sequentially selecting hyperparameters and greedily choosing the parameters that yielded the best metric values, was as follows: Solver: newton-cholesky; Tolerance: 1e-5; Maximum iterations: 250; With this configuration, the model achieved a accuracy of 81.29%±0.01% and quality on F1 metric of 72.69%±0.04%.

4.1.2. *Selection of optimal hyperparameters for gradient boosting.* The optimal configuration, determined by sequentially selecting hyperparameters and greedily choosing the parameters that yielded the best metric values, was as follows: Loss function: exponential; Learning rate: 1e-1; Number of estimators: 50; Criterion: friedman_mse; Maximum depth: 2; Minimum samples per leaf: 3. With this configuration, the model achieved a accuracy of 83.27%±0.01% and quality on F1 metric of 76.28%±0.42%.

4.1.3. *Selection of optimal hyperparameters for a neural network.* The optimal configuration, determined by sequentially selecting hyperparameters and greedily choosing the parameters that yielded the best metric values, was as follows: Number of hidden layers: 1; Hidden layer size: 7; Activation: ReLU Number of epochs: 600; Criterion: CrossEntropyLoss; Optimizer: Rprop. With this configuration, the model achieved a accuracy of 82.11%±6.2% and quality on F1 metric of 68.77%±4.12%. However, the quality of the neural network varied greatly over time, and stable performance during training was not consistently achieved.

[1] `https://gitlab.com/LukianovKirill/trustai_72.git`

Table 1. Detection model accuracy on shifted data. Features to be shifted were selected for different prediction models in order to maximally decrease their quality.

Algorithm	**1**	**3**	**5**	**7**
Shift = 5				
Logistic Regression	0 ± 0	8 ± 1	97 ± 2	100 ± 0
XGB	0 ± 0	63 ± 4	95 ± 1	100 ± 0
NN	0 ± 0	57 ± 12	97 ± 1	100 ± 0
Shift = 15				
Logistic Regression	0 ± 0	38 ± 1	100 ± 0	100 ± 0
XGB	0 ± 0	43 ± 7	97 ± 1	100 ± 0
NN	0 ± 0	71 ± 6	99 ± 0	100 ± 0
Shift = 30				
Logistic Regression	0 ± 0	64 ± 1	100 ± 0	100 ± 0
XGB	0 ± 0	83 ± 5	100 ± 0	100 ± 0
NN	0 ± 0	69 ± 10	100 ± 0	100 ± 0

4.2. Experiments with shifts. For the detection of covariate shift, 70,000 records of real disk data were used. Additionally, about 80,000 records were obtained when creating a grid around the real data.

For each experiment, the results were averaged over 25 runs. Each run included the following steps: model training, testing, selecting feature combinations with fixed shifts where the model quality decreased the most, conducting shift detection before and after feature shifts, and then removing the shift using a statistical method.

Results for the shift detection experiment are shown in Table 1. From the experiments, it can be inferred that the detection quality naturally increases with the number of features subjected to shifts and the magnitude of the shift. It was also noted that changing only one feature always resulted in extremely low detection quality. This can be explained by the fact that the feature that most often had the greatest influence on the models covered a wide range of values (from 20 to 100) without significant gaps. When the data were shifted based on this feature, many points changed class but did not leave the training point distribution.

Table 2. F1 quality of different models prediction on clean data, shifted data, and calibrated data after using a statistical method

Algorithm	Clean data	Shifted data	Data after calibration
Shift = 1 feature			
Logistic Regression	72.3	60.4	72
XGB	76	67.8	72.1
NN	66	44.7	66
Shift = 3 feature			
Logistic Regression	72.6	60.6	72.2
XGB	76.2	62.3	69.8
NN	67.8	44.8	67.7
Shift = 5 feature			
Logistic Regression	71.9	60.4	71.6
XGB	75.8	61.8	70.3
NN	67.8	45.1	66.1

The detection quality on unchanged data was consistently around 97–100%, so a separate graph for shift detection without any shift is not provided.

Table 2 shows the prediction results on clean data, after shifting, and after removing the shift using a statistical method. Since the overall trend remained consistent for different shift magnitudes, the results are presented only for a 15-unit shift. It can be observed that the quality decreases after shifting but rises after using the statistical method to remove the shift.

§5. Conclusion

In this work, we proposed a novel approach to training covariate shift detection models that does not rely on additional real data or artificial modeling of shifts. Moreover, we introduced a methodology for integrating shift detection, administrator warning, shift elimination and subsequent HDD failure prediction. Our proposed methods have demonstrated high efficacy of shift detection and elimination across various experiments, indicating their potential for real-world implementation.

Future research could explore automated techniques for removing covariate shifts, with a particular focus on methods such as additional model training, along with exploring the possibility of combining the proposed detection method with other techniques to enhance efficiency when shifting a small number of features.

References

1. P.P. Shinde and S. Shah, *A review of machine learning and deep learning applications*, in: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) (2018), pp. 1–6.
2. M. Sugiyama and A.J. Storkey, *Mixture regression for covariate shift*, in: Advances in Neural Information Processing Systems, vol. 19 (2006).
3. A. Bathelt, N.L. Ricker, and M. Jelali, *Revision of the Tennessee Eastman process model*. — IFAC-Pap. **48** (2015).
4. M. Kravchik and A. Shabtai, *Efficient cyber attack detection in industrial control systems using lightweight neural networks and PCA*. — IEEE Trans. Dependable Secure Comput. **19** (2021).
5. N.L. Ricker, *Decentralized control of the Tennessee Eastman challenge process*. — J. Process Control **6** (1996).
6. S. Yin, S.X. Ding, A. Haghani, H. Hao, and P. Zhang, *A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process*. — J. Process Control **22** (2012).
7. G. Bontempi, S. Ben Taieb, and Y.A. Le Borgne, *Machine learning strategies for time series forecasting*, in: Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures 2 (2013).
8. Z. Tao, Q. Xu, X. Liu, and J. Liu, *An integrated approach implementing sliding window and DTW distance for time series forecasting tasks*. — Appl. Intell. (2023).
9. E.G. Silva, P.S.G. De Mattos Neto, and G.D.C. Cavalcanti, *A dynamic predictor selection method based on recent temporal windows for time series forecasting*. — IEEE Access **9** (2021).
10. M. Sugiyama and M. Kawanabe, *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*, MIT Press, 2012.
11. M.D. Prieto, G. Cirrincione, A.G. Espinosa, J.A. Ortega, and H. Henao, *Bearing fault detection by a novel condition-monitoring scheme based on statistical-time features and neural networks*. — IEEE Trans. Ind. Electron. **60** (2012).
12. A.S. Raj and N. Murali, *Early classification of bearing faults using morphological operators and fuzzy inference*. — IEEE Trans. Ind. Electron. **60** (2012).
13. W.-Y. Lin, Y.-H. Hu, and C.-F. Tsai, *Machine learning in financial crisis prediction: a survey*. — IEEE Trans. Syst. Man Cybern. C **42** (2011).
14. K. Bluwstein, M. Buckmann, A. Joseph, S. Kapadia, and Ö. Śimşek, *Credit growth, the yield curve and financial crisis prediction: Evidence from a machine learning approach*. — J. Int. Econ. (2023).

15. J. Uthayakumar, N. Metawa, K. Shankar, and S.K. Lakshmanaprabu, *Intelligent hybrid model for financial crisis prediction using machine learning techniques.* — Inf. Syst. E-Bus. Manag. **18** (2020).
16. Z. Wang, X. Su, and Z. Ding, *Long-term traffic prediction based on LSTM encoder-decoder architecture.* — IEEE Trans. Intell. Transp. Syst. **22** (2020).
17. W. Wang and M. Carr, *A stochastic filtering-based data-driven approach for residual life prediction and condition-based maintenance decision-making support*, in: 2010 Prognostics and System Health Management Conference (2010), pp. 1–10.
18. P. Lall, R. Lowe, and K. Goebel, *Extended Kalman filter models and resistance spectroscopy for prognostication and health monitoring of lead-free electronics under vibration.* — IEEE Trans. Reliab. **61** (2012).
19. S. Oeyen, K. Vermeulen, D. Benoit, L. Annemans, and J. Decruyenaere, *Development of a prediction model for long-term quality of life in critically ill patients.* — J. Crit. Care **43** (2018).
20. G.F. Hughes, J.F. Murray, K. Kreutz-Delgado, and C. Elkan, *Improved disk-drive failure warnings.* — IEEE Trans. Reliab. **51** (2002).
21. V. Chandola, A. Banerjee, and V. Kumar, *Anomaly detection: A survey.* — ACM Comput. Surv. (CSUR) **41** (2009).
22. R. Chalapathy and S. Chawla, *Deep learning for anomaly detection: A survey*, ArXiv preprint arXiv:1901.03407 (2019).
23. D. Samariya and A. Thakkar, *A comprehensive survey of anomaly detection algorithms.* — Ann. Data Sci. **10** (2023).
24. F. Salfner, M. Lenk, and M. Malek, *A survey of online failure prediction methods.* — ACM Comput. Surv. (CSUR) **42** (2010).
25. J.F. Murray, G.F. Hughes, and K. Kreutz-Delgado, *Hard drive failure prediction using non-parametric statistical methods*, in: Proceedings of ICANN/ICONIP (2003).
26. Y. Wang, E.W.M. Ma, T.W.S. Chow, and K.-L. Tsui, *A two-step parametric method for failure prediction in hard disk drives.* — IEEE Trans. Ind. Inform. **10** (2013).
27. L.P. Queiroz, F.C.M. Rodrigues, J.P.P. Gomes, F.T. Brito, I.C. Chaves, M.R.P. Paula, M.R. Salvador, and J.C. Machado, *A fault detection method for hard disk drives based on mixture of Gaussians and nonparametric statistics.* — IEEE Trans. Ind. Inform. **13** (2016).
28. N. Georgoulopoulos, A. Hatzopoulos, K. Karamitsios, I.M. Tabakis, K. Kotrotsios, and A.I. Metsai, *A survey on hardware failure prediction of servers using machine learning and deep learning*, in: 2021 10th International Conference on Modern Circuits and Systems Technologies (MOCAST) (2021), pp. 1–5.
29. J. Li, R.J. Stones, G. Wang, X. Liu, Z. Li, and M. Xu, *Hard drive failure prediction using decision trees.* — Reliab. Eng. Syst. Saf. **164** (2017).
30. J. Li, X. Ji, Y. Jia, B. Zhu, G. Wang, Z. Li, and X. Liu, *Hard drive failure prediction using classification and regression trees*, in: 2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (2014), pp. 383–394.

31. Q. Li, H. Li, and K. Zhang, *Prediction of HDD failures by ensemble learning*, in: 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS) (2019), pp. 237–240.
32. Y. Ding, Y. Zhai, Y. Zhai, and J. Zhao, *Explore deep auto-coder and big data learning to hard drive failure prediction: a two-level semi-supervised model.* — Connect. Sci. **34** (2022).
33. R. Pinciroli, L. Yang, J. Alter, and E. Smirni, *Lifespan and failures of SSDs and HDDs: Similarities, differences, and prediction models.* — IEEE Trans. Dependable Secure Comput. (2021).
34. S. Zahariah and H. Midi, *Minimum regularized covariance determinant and principal component analysis-based method for the identification of high leverage points in high dimensional sparse data.* — J. Appl. Stat. **50** (2023).
35. Y. Wang, K.L. Tsui, E.W.M. Ma, and M. Pecht, *A fusion approach for anomaly detection in hard disk drives*, in: Proceedings of the IEEE 2012 Prognostics and System Health Management Conference (PHM-2012 Beijing) (2012), pp. 1–5.
36. S.M. Djurasevic, U.M. Pesovic, and B.S. Djordjevic, *Anomaly Detection Model for Predicting Hard Disk Drive Failures.* — Appl. Artif. Intell. **35** (2021).
37. M.W. Pozen, R.B. D'Agostino, H.P. Selker, P.A. Sytkowski, and W.B. Hood Jr, *A predictive instrument to improve coronary-care-unit admission practices in acute ischemic heart disease: a prospective multicenter clinical trial.* — N. Engl. J. Med. **310** (1984).
38. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, *LightGBM: A highly efficient gradient boosting decision tree*, in: Advances in Neural Information Processing Systems, vol. 30 (2017).
39. F. Rosenblatt, *The perceptron, a perceiving and recognizing automaton. Project Para*, Cornell Aeronautical Laboratory (1957).

Поступило 15 ноября 2024 г.

Ivannikov Institute for System Programming of the Russian Academy of Sciences;
Moscow Institute of Physics and Technology (National Research University);
ISP RAS Research Center for Trusted Artificial Intelligence, Moscow, Russia

E-mail: `lukyanov.k@ispras.ru`

Ivannikov Institute for System Programming of the Russian Academy of Sciences;
ISP RAS Research Center for Trusted Artificial Intelligence, Moscow, Russia

E-mail: `drobyshevsky@ispras.ru`

Ivannikov Institute for System Programming of the Russian Academy of Sciences;
ISP RAS Research Center for Trusted Artificial Intelligence, Moscow, Russia

E-mail: `turdakov@ispras.ru`