Maintaining and Monitoring AIOps Models Against Concept Drift

Poenaru-Olaru, Lorena; Cruz, Luis; Rellermeyer, Jan S.; Van Deursen, Arie

DOI
10.1109/CAIN58948.2023.00024

Publication date
2023

Document Version
Final published version

Published in
Proceedings - 2023 IEEE/ACM 2nd International Conference on AI Engineering - Software Engineering for AI, CAIN 2023

Citation (APA)
Poenaru-Olaru, L., Cruz, L., Rellermeyer, J. S., & Van Deursen, A. (2023). Maintaining and Monitoring AIOps Models Against Concept Drift. In *Proceedings - 2023 IEEE/ACM 2nd International Conference on AI Engineering - Software Engineering for AI, CAIN 2023* (pp. 98-99). (Proceedings - 2023 IEEE/ACM 2nd International Conference on AI Engineering - Software Engineering for AI, CAIN 2023). IEEE. https://doi.org/10.1109/CAIN58948.2023.00024

2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN) | 979-8-3503-0113-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/CAIN58948.2023.00024

Maintaining and Monitoring AIOps Models Against Concept Drift

Lorena Poenaru-Olaru*, Luis Cruz*, Jan S. Rellermeyer †, Arie van Deursen*
*Software Engineering, TU Delft, Netherlands
†Dependable and Scalable Software Systems, Leibniz University Hannover, Germany
L.Poenaru-Olaru@tudelft.nl, L.Cruz@tudelft.nl, rellermeyer@vss.uni-hannover.de, arie.vandeursen@tudelft.nl

***Abstract*—AIOps solutions enable faster discovery of failures in operational large-scale systems through machine learning models trained on operation data. These models become outdated during the occurrence of concept drift, a term used to describe shifts in data distributions. In operation data concept drift is inevitable and it impacts the performance of AIOps solutions over time. Therefore, concept drift should be closely monitored and immediate maintenance to prevent erroneous predictions is required. In this work, we propose an automated maintenance pipeline for AIOps models that monitors the occurrence of concept drift and chooses the most appropriate model retraining technique according to the drift type.**

***Index Terms*—machine learning model lifecycle, AIOps, concept drift detection, concept drift adaptation**

I. Introduction

Artificial Intelligence for IT operations (AIOps) enables the automation of traditional IT operation processes of large-scale software systems. Examples AIOps solutions are predicting node, job, or disc failure [5]. Not identifying these failures in time could lead to substantial monetary costs among enterprises. Thus, the correctness of AIOps models' prediction is crucial [4].

The quality of AIOps models' predictions is affected by the presence of *concept drift* in operation data. In general, concept drift leads to losses in deployed machine learning models' performance over time [7]. Such effects were observed in AIOps solutions for predicting job and disk failures [5].

The degradation in performance is a consequence of the evolving operation data, whereby the data learned by the model during the training process no longer resembles the data used to evaluate the model. The machine learning models used in AIOps solutions assume that the data distribution is not changing over time and, thus, the evaluation data should be similar to the training data. Therefore, evolving data is a violation of this assumption, and their performance is affected [7]. For this reason, the integration of machine learning solutions into software systems is still challenging.

Mitigating the effects of concept drift on operation data would allow faster integration of AIOps solutions into software systems [5]. Therefore, in this paper, we propose automated and resource-efficient solutions to monitor and maintain AIOps models against concept drift. Furthermore, we highlight important future research directions in automating the maintenance of these models in production.

II. Proposed Approach

In past AIOps studies, the model maintenance is done periodically [5], [6], which implies that the model is retrained regularly. In this work, we propose a maintenance pipeline that retrains a model only when concept drift is detected, avoiding unnecessary retraining. An overview of our approach is presented in Figure 1. The general AIOps modeling process was extracted from previous work [5] and modified to include drift monitoring and model maintenance based on drift identification. The upcoming data is constantly monitored against concept drift. If concept drift is detected, the model is retrained according to the drift type determined in the drift identification block. If no drift is signaled, no action is performed.

A. Monitoring Concept Drift

Concept drift monitoring is usually done through *concept drift detectors*, which are algorithms that capture the moment when data shifts occur [3].

a) Concept drift detectors: Authors of [7] compared two types of concept drift detectors, namely *Error Rate-Based (ERB)*, and *Data Distribution-Based (DDB)* detectors.

The ERB detectors require the predicted labels of an AIOps model and the true labels to compute the error rate. The error rate is constantly measured and a significant change indicates a drift. These detectors are label-dependent since they assume the true labels' immediate availability. Obtaining true high-quality labels in AIOps implies tremendous annotation effort and costs [2]. Therefore, the adoption of these detectors to monitor operation data using true labels is impractical.

DDB detectors identify concept drift by assessing the discrepancies between the distributions of the training data and the distribution of the evaluation data. Thereby, unlike ERB detectors, they are not label-dependent, which makes them more suitable for monitoring concept drift in production.

b) Future Research Paths: The study of Lorena et al. [7] shows that, for some datasets, the class imbalance significantly impacted the accuracy of concept drift detectors. In operation data class imbalance is inevitable. Therefore, understanding whether drift detectors could be used to monitor concept drift in such situations requires further exploration. Thus, we need to understand how DDB detectors manage to detect drift in operation data since they are the most practical to use. Therefore, the research direction can be formulated as follows:

979-8-3503-0113-7/23/$31.00 ©2023 IEEE
DOI 10.1109/CAIN58948.2023.00024

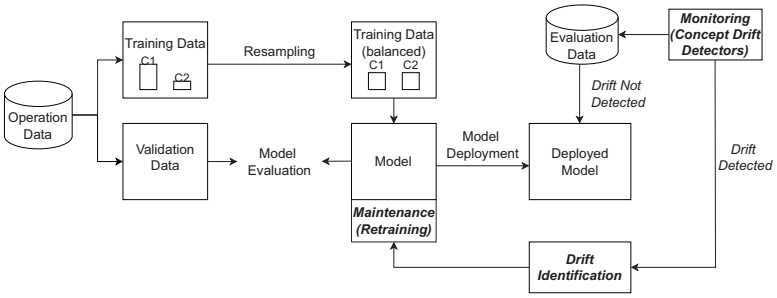

Fig. 1. Monitoring and Maintenance of AIOps Solutions Pipeline.

Analysing the drift detection accuracy of DDB detectors on operation data.

In most cases, the ERB detectors outperform the DDB detectors according to [7]. However, ERB detectors suffer from a high dependency on true labels, which for operation data are expensive to obtain. In [1] it is mentioned that some of these detectors were modified to estimate a pseudo-error rate instead of the true error rate, eliminating the necessity for true labels. To assess their suitability for AIOps models further research is required to understand whether the unavailability of true labels influences the drift detection performance. Thus we can summarize the research path as follows:

Analyzing the drift detection accuracy of ERB detectors used in an unsupervised fashion on operation data.

B. Maintenance Against Concept Drift

Maintaining machine learning models in an environment where the data is constantly changing over time is done through concept drift adaptation techniques [3].

a) Concept Drift Adaptation Techniques: The most commonly used adaptation technique is *model retraining* [5]. There are two state-of-the-art alternatives to retrain a model [6], the *sliding window approach* and the *full history approach* [6].

The *sliding window approach* updates the model by retraining it on the most recent data. Thus, the old data is discarded and the assumption is that the recent data is the most similar to the data the model will be evaluated on. The *full history approach* retrains the model using all the available labeled data during the model update. Thereby, the assumption is that more data helps the model to generalize better.

b) Future Research Paths: According to [5] both the sliding window and the full history approach lead to good results. However, according to the situation, one approach might result in better model performance. Sometimes the drift is seasonal and, thereby, observed periodically. For instance, the usage of some servers might increase during specific periods, which could result in more job failures. In this case, the full history approach includes data that captures multiple such periods compared to the sliding window approach, which discards this data. Thus, the full history technique should be used for this situation. However, the sliding window technique has the advantage that the model solely learns from the new data, which is the most similar to the upcoming data. For instance, this technique is suitable when only the data collected after a software update should be considered.

Deriving an automated and systematic maintenance process requires an in-depth understanding of which retraining technique to use in which situation. Thereby, we propose a drift identification step, which is able to detect the drift type. Furthermore, we need more understanding about which retraining techniques are preferred over the others and in which situations. We propose the following research path:

Identifying the drift type and the most suitable retraining technique according to the drift type in operation data.

III. Conclusions

In this paper, we discuss techniques to overcome the impact of concept drift on the AIOps models' performance over time. We further propose a model maintenance pipeline involving two main parts, namely monitoring concept drift and maintenance against concept drift. The former identifies evaluation data batches with concept drift and the latter updates the model with the most appropriate retraining technique according to the drift type. Furthermore, we present the main research paths derived from having an automated system to maintain AIOps in production, enabling faster AIOps integration.

References

[1] Bayram, F., Ahmed, B.S., Kassler, A.: From concept drift to model degradation: An overview on performance-aware drift detectors. Knowl. Based Syst. **245**, 108632 (2022)

[2] Dang, Y., Lin, Q., Huang, P.: Aiops: Real-world challenges and research innovations. In: 2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion). pp. 4–5 (2019)

[3] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., Bouchachia, A.: A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)* 46, 1 – 37 (2014)

[4] Li, Y., Jiang, Z.M.J., Li, H., Hassan, A.E., He, C., Huang, R., Zeng, Z., Wang, M., Chen, P.: Predicting node failures in an ultra-large-scale cloud computing platform: An aiops solution. ACM Trans. Softw. Eng. Methodol. **29**(2) (apr 2020)

[5] Lyu, Y., Li, H., Sayagh, M., Jiang, Z.M.J., Hassan, A.E.: An empirical study of the impact of data splitting decisions on the performance of aiops solutions. ACM Trans. Softw. Eng. Methodol. **30**(4) (2021)

[6] Lyu, Y., Rajbahadur, G.K., Lin, D., Chen, B., Jiang, Z.M.J.: Towards a consistent interpretation of aiops models. ACM Trans. Softw. Eng. Methodol. **31**(1) (nov 2021)

[7] Poenaru-Olaru, L., Cruz, L., van Deursen, A., Rellermeyer, J.S.: Are concept drift detectors reliable alarming systems? - a comparative study. In: 7th Workshop on Real-time Stream Analytics, Stream Mining, CER/CEP & Stream Data Management in Big Data (2022)