KPI Data Anomaly Detection Strategy for Intelligent Operation and Maintenance Under Cloud Environment

Youchang Xu, Ningjiang Chen, Ruwei Huang, Hanlin Zhang

KPI Data Anomaly Detection Strategy For Intelligent Operation and Maintenance Under Cloud Environment

Youchang Xu*, Ningjiang Chen*, Ruwei Huang*,Hanlin Zhang*

*School of Computer and Electronic Information, GuangXi University
53000, Nanning, China
chnj@gxu.edu.cn

ABSTRACT. In the complex and changeable cloud environment, monitoring and anomaly detection of the cloud platform is very important. In the cloud environment, because of the complex structure of the system, the characteristics of the monitoring data are constantly changing. In order to adapt to the change of the data characteristics, the operators need to adjust the anomaly detection model to solve the problem of dynamic KPI anomaly detection, this paper transforms the adjustment process of anomaly detection model into a general Markov decision process by means of reinforcement learning technology, which cloud reduce the human cost caused by anomaly detection model adjustment, and improve the effective detection rate of the anomaly detection model. Comparing the three typical KPI curves with other optimization strategies, and finally verify the effectiveness of the strategy used in this paper.

Keywords: anomaly detection; Markov decision process; automatic parameter adjustment

1 Introduction

With the development of cloud computing technology, most enterprises move services into the cloud to achieve better performance and security. In order to ensure the reliability and stability of the cloud platform, the operator obtain a large number

of monitoring data from different levels of the cloud platform forming a real-time KPI (Key Performance Indicator) curve to observe the running state of the key components of the cloud platform and using anomaly detection model to analyse historical KPI data build a prediction model of KPI curve under normal conditions. In practice, enterprise will formulate marketing strategy according to the market, which will result in the change of data characteristics of the KPI curve which is monitored in the cloud, makes the anomaly detection model cause a lot of false alarm and form the alarm storm.

In order to solve the problem of above, this paper proposes an intelligent KPI data anomaly detection strategy. Firstly, differential and autocorrelation functions are used to construct a perceptron for the change of KPI data characteristics.Then, in the face of the changes in KPI data features, this paper builds an intelligent regulator of time series model based on reinforcement learning technology, and gets rid of the dependence on manual and label data. At the end of this paper, three typical KPI data in the real environment are detected. The experimental results show that the proposed method can adapt to the constant change of the KPI data characteristics under the cloud environment, accurately judge the characteristics of the KPI curve, and intelligently adjust the corresponding anomaly detection algorithm to ensure the effective anomaly detection of the cloud platform.

The remaining part of this paper is organized as follows. The second chapter introduces the whole idea of this article. The third chapter introduces the intelligent anomaly detection framework constructed in this paper. The fourth chapter introduces the related anomaly detection contrast experiments on three typical KPI data, and the fifth chapter introduces the related work with this paper. The sixth chapter summarizes the work and prospects for the future.

2 The core idea

In the cloud environment, the change of the KPI data features used to describe the running state of the system modules is uncontrollable. In order to realize automatic anomaly detection of KPI curves in cloud environment, the data features of KPI curves need to be perceived first. This paper compares the global trend of KPI data with the local trend by using the differential technique and autocorrelation function, thus

perceiving whether the data characteristics of the KPI curve have changed. Different time series algorithms have different parameters range, therefore, it is very important to abstract the algorithm of different time series into a unified mode of adjustment. In this paper, we use the reinforcement learning technology to transform the adjustment process of different time series algorithms into the process of finding the optimal parameters. Because the reinforcement learning technology can interact with the environment to produce data, it can get rid of the dependence on the historical data and quickly adjust the anomaly detection model. In this paper, an anomaly detection framework for intelligent operation and maintenance is constructed, as shown in Figure 1.

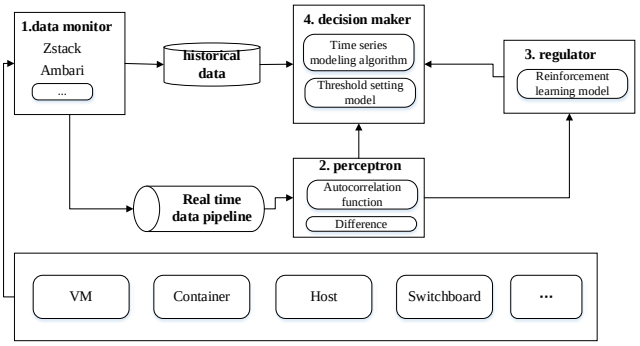


Figure 1. *Anomaly detection framework*

Specific modules are (1) **Data monitor**: it is different cloud products to collect KPI data at different levels in the cloud environment and carry out persistent storage;(2) **Perceptron**: compare the global fluctuation trend of KPI data with the local fluctuation trend by using the difference technique and autocorrelation function to judge whether the KPI data characteristics of the real-time flow change;(3) **Adjuster**: Based on the interaction between the reinforcement learning technology and the external environment, the adjustment of model of different time series is transformed into an automated Markov decision process, which makes the adjustment process free from the manual participation and the self-healing recovery;(4) **Decision maker**: using a variety of time series algorithms to predict the KPI curve, by comparing the relative deviations between the true value and the predicted value, determine whether the KPI curve is anomaly in the set threshold.

3 Automatic anomaly detection method

3.1 KPI data feature perception

KPI data is essentially a continuous time series data, The characteristics of data are periodic, stable and unstable. On the periodic determination, for the monitoring data set DS, this paper uses the differential technique to carry out the difference processing to the global data and compare the changes of the global variance before and after the difference. If the monitoring data set is periodic, the global variance $V_g(DS)$ before the difference will be far greater than the global variance $V_g'\big(diff(DS)\big)$ after the difference, so the use of Formula 1 can determine whether the monitoring data set is periodic.

$$\frac{V_g(DS)}{V_g'\big(diff(DS)\big)+V_g(DS)} = 1 \tag{1}$$

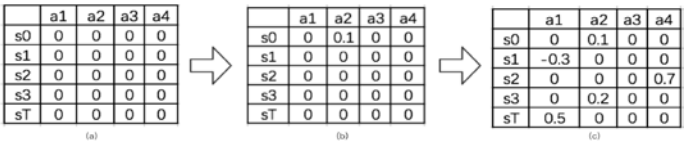
On the determination of stable and unstable, we calculate the autocorrelation function$\widehat{\rho_k}$ of monitoring data set, such as formula 2,

$$\widehat{\rho_k} = \frac{\sum_{t=1}^{T-k}(p_t-\bar{p})(p_{t+k}-\bar{p})}{\sum_{t=1}^{T}(p_t-\bar{p})^2} \tag{2}$$

It can identify whether the time series data have stability, if the autocorrelation function of the KPI curve does not decrease rapidly with the change of the adjacent time points to 0, then the KPI curve has unstable and vice versa.

3.2 Automatic adjustment of time series model

The Q-Learning [11] algorithm is the main method to solve the model free reinforcement learning. Its basic idea is to record the utility value of the state in each action, that is, the action state value, by establishing a function table. The action state value represents the validity and value of the action selected under the current state, and also as the basis for the next strategy to select the action, and updates the action state value of the current state through the action state value of the next state, as shown in Figure 2(the data in the diagram is used for demonstration):



	a1	a2	a3	a4
s0	0	0	0	0
s1	0	0	0	0
s2	0	0	0	0
s3	0	0	0	0
sT	0	0	0	0

(a)

	a1	a2	a3	a4
s0	0	0.1	0	0
s1	0	0	0	0
s2	0	0	0	0
s3	0	0	0	0
sT	0	0	0	0

(b)

	a1	a2	a3	a4
s0	0	0.1	0	0
s1	-0.3	0	0	0
s2	0	0	0	0.7
s3	0	0.2	0	0
sT	0.5	0	0	0

(c)

Figure 2. *$Q(s,a)$ function*

The initial value of function table $\mathrm{Q(s,a)}$ is (a). In one strategy, s_0 is selected randomly from the action of non-negative value in the initial state, In Figure 2 (b), a_2 is selected so that the state becomes S1, and the utility value is 0.1 by formula 3, where r is the immediate reward given by the reward function, $Q(s_{t+1}, a_{t+1})$ is the utility value of the next state, 0 in Figure 2, and the update function table as shown in Figure 2 (c) at the end of a strategy.

$$Q(s_t, a_t) = r + \gamma(\max(Q(s_{t+1}, a_{t+1}))) \tag{3}$$

In the process of action selection, the Q-Learning algorithm is selected according to the non-negative value of the corresponding $\mathrm{Q(s,a)}$function table, such as the next policy, figure 2 (c) as the basis, the optional action at $\mathrm{s_0}$ is $\{\mathrm{a_1, a_2, a_3, a_4}\}$, and the optional action at $\mathrm{s_1}$ is $\{\mathrm{a_2, a_3, a_4}\}$, The execution of each strategy will update the $\mathrm{Q(s,a)}$ function table until the $\mathrm{Q(s,a)}$ function table converge as Figure 3, and at this time the optimal strategy is selected to select the maximum cumulative return value, that is, the maximum utility value for each pair state-action is selected, for example, the optimal strategy in Figure 3 is a sequence $\tau = \{a_4, a_2, a_4, a_1, a_1\}$.

	a1	a2	a3	a4
s0	0.2	0.1	-0.4	0.3
s1	-0.3	0.6	-0.5	0.2
s2	0.3	-0.4	-0.6	0.7
s3	0.75	0.2	-0.6	0.3
sT	0.5	-0.4	0.4	0.2

Figure 3. *Convergent function table $Q(s,a)$*

In the Q-Learning algorithm, the setting of the reward function is static. It gives rewards or penalties depending on whether the current action makes the model state better than the initial state or whether it is superior to the previous state. But at this time, there will be a state of *S1* in Figure 3. When the function table is not yet convergent, there are always multiple actions to choose from in this state. Most actions do not bring optimization to the current model, which makes many invalid iteration steps in finding optimal strategies. We want to reduce the steps that can't make the

model state transition to better in the overall adjustment process, so that the function table can converge faster, so this article set the dynamic return function as formula 4:

$$R = \frac{F_t}{F_T} \cdot (F_t - F_{max}) \quad (4)$$

F_t is the F-Score value obtained from the anomaly detection model under the current parameter adjustment action, that is, the current state value. F_T is the target state, $\frac{F_t}{F_T}$ makes the current state value closer to the target value, and the bigger the reward value is, F_{max} is the maximum state value set during the execution of a policy. $F_t - F_{max}$ makes the award be rewarded only if the exception detection model gets better state values under the adjustment action, otherwise it will be punished, which is beneficial to a strategy to reach the optimal state faster. Based on the above strategy, we get the pseudo code for obtaining the best policy based on Q-Learning algorithm, as shown in Table 1:

Table.1. Optimal strategy seeking algorithm

Optimal strategy seeking algorithm based on Q-Learning
Input: Initialization parameters (x, y)
1:$A = \{x, x + p(x), x - p(x)\} \times \{y, y + p(y), y - p(y)\}$
2:**Initialization** $Q(s, a), \forall s \in S, a \in A(s)$, Given parameter a, γ
3:**Repeat**:
4: Given initial state s, Choose action a according to ε greed strategy
5: Computational anomaly detection model score F_{max}
6: **Repeat**(episode):
7: Select action a according to $Q(s_t, a_t)$ in the state s
8: $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + a[R_t + \gamma max_a Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$
9: Calculation of the current anomaly model score F_t
10: $s_t \leftarrow s_{t+1}; a_t \leftarrow a_{t+1}$
11: $R_t = (F_t - F_{max}) \cdot R_t$
12: **IF** $R_t > 0$
13: $F_{max} = F_t$
14: **until** s is the terminated state
15: **until** all $Q(s, a)$ convergence
16:**output**: $\pi(s) = argmax_a Q(s, a)$

Through algorithm above, we compare the action selection process of static reward function, as shown in Figure4.

	a1	a2	a3	a4
s0	0	0.1	0	0
s1	-0.3	0	0	0
s2	0	0	0	0.1
s3	0	0.2	0	0
sT	0.5	0	0	0

(a)

	a1	a2	a3	a4
s0	0	0.1	0	0
s1	-0.3	0	0	0
s2	0	0	0	-0.2
s3	0	0.1	0	0
sT	0.3	0	0	0

(b)

Figure 4. *$Q(s,a)$ table comparison*

When the state s_1 is updated to s_2 in Figure 4 (a), according to the static reward function, action a_4 makes the state of the model better than the previous state, so we get the reward. However, because the state a is punished, the whole model is not optimized. According to the dynamic reward function presented in this paper, we will get the punishment, as shown in Figure 4 (b). In the next policy execution, there are 4 actions that can be attempted in the state *S2* (a) table, while there are only 3 of them in (b). Therefore, (b) table will arrive at the next state faster, and this advantage will be more obvious in the accumulation of multiple strategies. Comparing the convergent function table $\mathrm{Q(s,a)}$ in Figure 3. Under the proposed strategy, the function table $\mathrm{Q(s,a)}$will converge as shown in Figure 5.

	a1	a2	a3	a4
s0	-0.2	0.7	-0.7	-0.3
s1	0.3	-0.2	-0.1	-0.5
s2	-0.2	0.5	-0.8	-0.6
s3	0	-0.1	-0.3	-0.4
sT	-0.2	0.5	-0.1	-0.8

Figure 5. *The function table $Q(s,a)$ of convergence under the dynamic reward function*

Compared to the static reward function, the dynamic reward function is stricter in the selection of the best strategy, so the utility value is more negative, and the adjustment action corresponding to the negative value will not be selected again, so the function table $\mathrm{Q(s,a)}$ will converge faster. Each line of the function table has at least one non- negative value, and it does not appear in a state without the optional action of a adjustment action. In experiment, we verified by experiments that the convergence speed of function table $\mathrm{Q(s,a)}$ is faster under dynamic reward function.

4 Experiment

4.1 Experimental design

In order to verify the effectiveness of the proposed strategy, this paper carries out the related experiments on the open desensitization data set [10] in the real environment of the Baidu Inc search data center. The physical environment of this experiment is 6 servers with 8 Cores CPU 32GB Mem. The programming environment is Anaconda 3.6. The comparison object is supervised learning strategy decision tree, unsupervised learning clustering strategy K-means, and parameter selection and estimation strategy in document [3]. In order to verify the effectiveness of the strategy in the anomaly detection, we observe and analyze the change process of the recall and accuracy of the anomaly detection results and compare the F-Score values of the different anomaly detection models. In order to verify the optimization of the strategy in the iterative process, the number of iterations per adjustment process is compared with the original Q-learning algorithm.

4.2 Evaluation index

(1) recall: the ratio of true outliers representing the true outliers detected by the representative, as shown in the formula 5.

$$\text{recall} = \frac{\#\ of\ ture\ anomalous\ points\ detected}{\#\ of\ ture\ anomalous\ points} \tag{5}$$

(2) precision: the ratio of the true outliers represented by the detection to the total outliers is calculated, as shown in 6.

$$precision = \frac{\#\ of\ ture\ anomalous\ points\ detected}{\#\ of\ anomalous\ points\ detected} \tag{6}$$

(3) F-Score: a comprehensive measure of recall and precision. The formula is shown in 7:

$$\text{F} - Score = \frac{2 \cdot recall \cdot precision}{recall + precision} \tag{7}$$

4.3 Experimental results

4.3.1 Verification of anomaly detection effect

. In this strategy, we set up F-Score > 0.70, F-Score > 0.80 as the target state of the anomaly detection model, select Holt-winters, ARIMA, EWMA, Wavelet as the model of time series, and take 1 days as the measurement and decision tree algorithm, clustering algorithm, parameter selection and estimation method in the 30 day data. Rate, accuracy and overall F-score value are obtained.

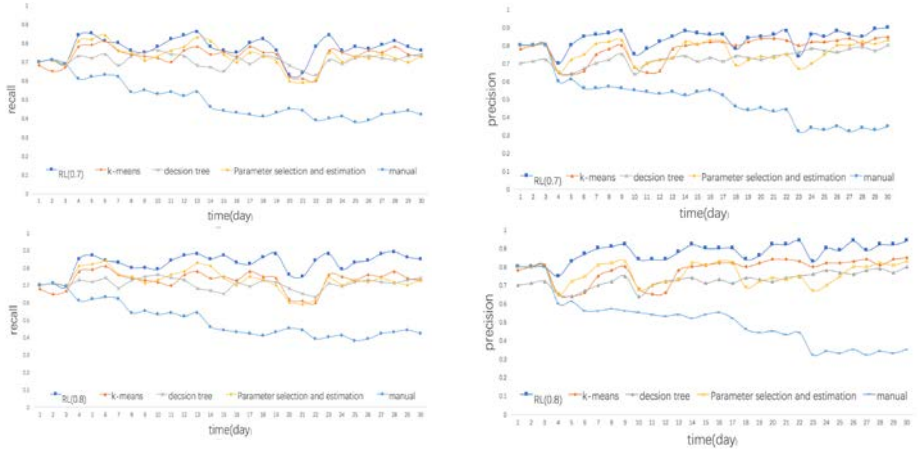


Figure 6 *Comparison of recall and precision*

From figure 6 , it can be seen that in the process of anomaly detection, the recall and the precision of the manual adjustment method when each data characteristic changes are further reduced. And other strategies can maintain good anomaly detection after adjustment. On the recovery of anomaly detection model adjustment, from the fourth day, the tenth day, the eighteenth day, the twenty-third day and the twenty-eighth day of figure 6, the strategy (RL) and parameter selection method proposed in this paper adjust the recovery fastest in the face of changes in data characteristics. Although both the decision tree strategy and the k-means method have high reliance on the characteristics of the new data, the decision tree strategy utilizes the markup update of the expert system, so that it is faster than the adjustment of the anomaly detection effect by the k-means. In the overall detection effect, the overall trend of the decision tree strategy shown in figure 6 is relatively stable, but the overall average is low. The overall fluctuations of RL, k-means, and parameter selection methods are large, but the overall average is high.

4.3.2 Optimization verification of iterative process

. In order to compare the number of iterations of the original Q-Learning algorithm and the optimized algorithm of this article when adjusting the parameters, we add the

counter of iterations in the iteration process to record the number of iterations. After the anomaly detection process, the data is obtained as shown in figure 7:

Figure7 *Iteration number comparison*

The original Q-Learning algorithm rewards each positive parameter adjustment action. The dynamic reward function proposed in this paper only rewards the current optimal parameter adjustment action, so that the action to obtain the reward is reduced, and the number of optional adjustment actions in the next adjustment is also reduced. As can be seen from figure 7 , five anomaly detection model adjustments are made in the anomaly detection process. In each adjustment, the optimized strategy of this paper is less than the original Q-Learning algorithm in the number of iterations. Therefore, the effectiveness of the proposed strategy for the optimization of the iterative process to obtain the best parameters is verified.

5 Related work

In cloud environment, many researchers have done research on anomaly detection algorithm. Some based on the data distribution to detect the anomaly, which using the inconsistency test method to compare the probability distribution of the detected data to the presumed probability distribution, such as the literature [1], and some methods based on deviation, such as ARIMA algorithm in literature [2], Holt-Winters algorithm in literature [3], Wavelet algorithm in literature [5]. However, these algorithms do not have a good solution to the change of data characteristics, and only rely on manual re-adjustment to achieve the desired detection efficiency.

To solve the problem of data characteristics over changing, researchers have made a study on the adaptive detection model. Some based on supervised learning technology, such as literature [8,9], Some based on unsupervised learning methods, such as literature [6,7], but those kind of algorithm usually needs to build an extra expert system to mark anomaly data, and has a high dependence on historical data. In literature [4], two strategies are used in parameter configuration. One is to enumerate the limited parameters by using the reduced parameter sample space and enumerate the spare parameters in advance. The other is to use the targeted parameter estimation algorithm to get the appropriate parameters. However, this method can’t guarantee that the reduced sample space contains the optimal parameters under each data characteristics in the pre-proposed parameter sample space, and for the complex anomaly detection algorithm, the corresponding parameter estimation method should be tested for each anomaly detection algorithm.

Based on the thought of the above work, this paper constructs an adaptive anomaly detection framework using the reinforcement learning technology, automatically triggering the adjustment of the anomaly detection model to the Markov decision process by perceiving the changes of the data characteristics, in addition, the strategy of selecting parameter adjustment action for different anomaly detection algorithms, which realizes the automatic adjustment of the anomaly detection model in the face of the change of data characteristics, and ensures a good anomaly detection effect in the cloud environment.

6 Conclusion

Anomaly detection is an important technology to ensure the stability of the system services of the cloud platform. However, because of the complexity of the data changes in the cloud environment, the anomaly detection model needs to be constantly adjusted. In this paper, we introduce an adaptive detection method based on reinforcement learning, which automatically triggers the transformation of the anomaly detection model to the Markov decision process by perceiving the changes in the characteristics of the monitoring data, and we put forward the selection strategy of parameter adjustment action and the optimization algorithm for obtaining the best parameters, and realize the automatic adjustment of the anomaly detection model when the data characteristics is changed. In the future work, we will further optimize the iterative process of the parameters of the Markov decision process, reduce the time of the parameter selection process, and improve the adaptability and sensitivity of the model in the anomaly detection process.

ACKNOWLEDGMENT

This work is supported by the Natural Science Foundation of China (No. 61762008, 61363003), Natural Science Foundation Project of Guangxi(No. 2017GXNSFAA198141), Key R&D project of Guangxi(No. GuiKE AB17195014), and R&D Project of Nanning(No. 20173161).

Reference

1. Ghanbari M, Kinsner W, Ferens K. Anomaly detection in a smart grid using wavelet transform, variance fractal dimension and an artificial neural network[C]// Electrical Power and Energy Conference. IEEE, 2016:1-6.
2. Laptev N, Amizadeh S, Flint I. *Generic and Scalable Framework for Automated Time-series Anomaly Detection*[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015:1939-1947.
3. Yang Y M, Yu H, Sun Z. A*ircraft failure rate forecasting method based on Holt-Winters seasonal model[C]//* IEEE, International Conference on Cloud Computing and Big Data Analysis. IEEE, 2017.
4. Liu D, Zhao Y, Xu H, et al. *Opprentice: Towards Practical and Automatic Anomaly Detection Through Machine Learning*[C]// Internet Measurement Conference. ACM, 2015:211-224.

5. Ge Z, Ge Z, Ge Z, et al. *G-RCA: a generic root cause analysis platform for service quality management in large IP networks*[C]// International Conference. ACM, 2010:5.
6. Chaaya G, Maalouf H. *Anomaly detection on a real-time server using decision trees step by step procedure*[C]// International Conference on Information Technology. 2017:127-133.
7. Zeb K, Assadhan B, Al-Muhtadi J, et al. *Anomaly detection using Wavelet-based estimation of LRD in packet and byte count of control traffic*[C]// International Conference on Information and Communication Systems. IEEE, 2016.
8. De Nadai M, Van Someren M. *Short-term anomaly detection in gas consumption through ARIMA and Artificial Neural Network forecast*[C]// Environmental, Energy and Structural Monitoring Systems. IEEE, 2015:250-255.
9. Hirata T, Kuremoto T, Obayashi M, et al. *Time Series Prediction Using DBN and ARIMA*[C]// International Conference on Computer Application Technologies. IEEE, 2016:24-29.
10. *https://github.com/baidu/Curve*
11. Aksaray D, Jones A, Kong Z, et al. *Q-Learning for robust satisfaction of signal temporal logic specifications*[C]// Decision and Control. IEEE, 2016.