Business Anomaly Detection Based on QoS Benchmark of Resource-service Chain in Collaborative Task

Haibo Li*, Juncheng Tong, Zheng Zhang and Yongbo Yu

1College of Computer Science and Technology, Huaqiao University, Xiamen, Fujian, 361021, China
*Corresponding author’s e-mail: lihaibo@hqu.edu.cn

Abstract. To detect business anomaly caused by resource services in collaborative task, an approach based on QoS benchmark of resource-service chain is presented. Firstly, taking QoS of resource services as feature, the QoS benchmarks are obtained by clustering the relationships between the features of resource services in resource service chains. Then, thresholds of QoS benchmarks are calculated. A dynamic calibration strategy to the threshold is used to perform online detection of business anomaly in global and local views of a collaborative task. Finally, experiments and case studies show that the proposed approach has a high accuracy rate for business anomaly detection.

1. Introduction

During a collaborative task, business delay or failure often occurs because of various business anomalies caused by too many unpredictable factors. Capturing the states of business anomalies as soon as possible can ensure the successful execution of a business process and improve the market competitiveness of an enterprise. Most business anomalies are caused by improper use of resource services, including resource conflicts, violation of constraints, etc. The value of QoS (Quality of Service) is usually used to measure the level of service satisfaction. Business anomaly occurs when some values of QoS have changed abruptly, such as significant increase in the cost of services and a noticeable decrease in passing rates of a product. Resource services used in sequential order are called the Resource-Service Chain (RSC). In an RSC, the relationship of QoS values between resource services should be discovered.

At present, a variety of methods for anomaly detection have been proposed. The threshold-based business anomaly detection method is one of the commonly used methods [1]. The thresholds of different indicators are set and taken as the standard of business anomaly detection [2-4]. If the corresponding service data is not within the threshold, there is anomaly in the current business. In these methods of threshold-based business anomaly detection, determining a reasonable threshold is a key problem [5]. Dynamic threshold and self-adaptation threshold are used in business anomaly detection methods [6-7]. However, deviation may still result due to subjective judgements depending on experience. In addition, business anomaly detection depends on only individual activities but also interoperability among the activities in a collaborative business process. Therefore, indicators of individual activities are difficult to reflect the whole of collaborative task.

In summary, a method of business anomaly detection based on RSC QoS benchmark is proposed in this paper, which is called QoSB_AD (QoS Benchmark-based Anomaly Detection). In QoSB_AD, the QoS attributes of resource services are used to mining the QoS benchmark of an RSC. The interoperability among the resource services is calculated via the values of QoS. Through dynamically
adjusting threshold in real time, the accuracy of the business anomaly detection in the collaborative
task can be improved.

The rest of the article is organized as follows. In Section 2, we proposed some concepts and
mathematical description to describe the problem of business anomaly detection. In Section 3, a
business anomaly detection method based on QoS benchmark of resource service chain is presented.
The design of the experiments and the results evaluation are presented in Section 4. Finally, in Section
5, a summary and a discussion of the future work are presented.

2. Problem definition

If some resource services are invoked by business activities in a certain sequence, they form a chain of
resource services, called the Resource-Service Chain (RSC) [8-9]. The RSC is defined as follows.

**Definition 1** Resource Service Chain (RSC), which can be expressed as $RSC = \{<RS_1, RS_2, ..., RS_n> | n \leq Activity\}$, where $RS_1, RS_2, ..., RS_n$ represent the resource service set selected by the business
activity. Among them, $Activity$ represents all business activities in the business process, and $RS$
represents the collection of resources selected when the business process is executed. The resource
service chain instance corresponding to the $RSC$ is formally defined as $rsc = \{<rs_1, rs_2, ..., rs_n> | n \leq Activity\}$.

The services resources provide can be measured by the values of QoS. Therefore, by mining the
QoS in business data, a series of benchmark can be established.

**Definition 2** RSC QoS Benchmark, which can be expressed as $QoSB = \{<Index_1, Index_2, ..., Index_n> | n \leq Activity\}$, where $Index_i$ represents the QoS value set of the resource services $RS_i$ in an $RSC$.

In business data, the execution of each business process and the resource services are recorded. By
analyzing the RSC QoS Benchmark and mining the business data, a reliable detection threshold can be
established.

When detecting a business anomaly for a collaborative task, it is necessary to set an appropriate
threshold to RSC QoS Benchmark. The threshold can be dynamically adjusted.

**Definition 3** Business Anomaly (BA) can be expressed as $BA = \{aid, rsid, RS, Threshold, AbnormalIndex\}$, where $Threshold$={$\delta_{aij}, \delta_{aij}, \delta_{aij}, \delta_{aij}, \delta_{aij}$}, $\Delta_{aij}$ represents the threshold set corresponding to the RSC QoS, $\Delta_{aij}$ represents the threshold of the RSC QoS similarity, $i, j \geq 1$; AbnormalIndex is the QoS value set of the resource service deviation, for each $index_k$ belong to
AbnormalIndex. If $threshold_k$ is the threshold corresponding to $index_k$, the constraint $threshold_k > index_k$ or $threshold_k < index_k$ must be satisfied.

The problem we want to solve in this paper is to calculate the AbnormalIndex for an RSC in
collaborative tasks.

3. Method of business activity anomaly detection

3.1 Clustering RSC QoS into benchmark

Generally, different categories of resource services have different QoS evaluation criteria. Therefore, it
is more practical to compare the similarity of QoS between resource services in same category than in
different one. Let $k$ be the number of QoS indicators in a category of resource service. $V(q_{aj})$ and $V(q_{bj})$
respectively represent the normalized QoS values of resource service $rs_a$ and $rs_b$. We use the following
formula to measure the similarity of $rs_a$ and $rs_b$.

$$distr(rs_a, rs_b) = \left(\sum_{j=1}^{k} \left(V(q_{aj}) - V(q_{bj})\right)^2\right)^{1/2}$$  \hspace{1cm} (1)

The similarity between RSC instances is calculated by the following formula.

$$dist(rsc_a, rsc_m) = \max \left(\{distr(rs_{aj}, rs_{mj}) \ast \sum_{j=1}^{l} distr(rs_{aj}, rs_{mj})\} \right)$$  \hspace{1cm} (2)
In formula (2), $r_{sc_a}$ and $r_{sc_m}$ represent two RSC instances, $l$ represent the length of the RSC, and $r_{sa_j}$ and $r_{sa_j'}$ represent the resource services in $r_{sc_a}$ and $r_{sc_m}$.

After calculating the similarity between any two RSCs, the $k$-means algorithm is used to cluster all the RSCs into $k$ cluster centers. The QoS values of the RSCs corresponding to cluster centers are QoS benchmarks.

3.2 Business activity anomaly detection based on QoS threshold

Detecting business anomaly is to calculate both the QoS values of resource services and their QoS benchmarks. Two QoS thresholds should be calculated as follows.

1. Thresholds of QoS values. Let $\delta_{aij}$ be the lower bound of the $j$-th QoS values, belonging to resource service $r_{sa_i}$ used by business activity $a$, $r_{sa_i}\in \text{instanceof}(RS_i)$. Let $\delta_{aij}'$ be the corresponding upper bound of the thresholds. $V(q_{ij})$ represents the normalized value of the $j$-th QoS values. The threshold lower and upper bound of the QoS index value can be expressed by formula (3) and (4).

$$\delta_{aij} = \min_i(V(q_{ij}))$$ (3)

$$\delta_{aij}' = \max_i(V(q_{ij}))$$ (4)

2. Thresholds of similarity to the QoS benchmark. Let $A_{ai}$ be the threshold of the similarity between the resource service $r_s$ invoked by the service activity $a$ and the corresponding RSC QoS benchmark. In combination with formula (1), threshold can be established as formula (5).

$$A_{ai} = \max_{i}(\text{distr}(r_{sa_i},r_s))$$ (5)

The threshold of anomaly detection should be changed for dynamic adjustment because the QoS values of service resources continuously change. There are two methods to dynamically adjust the threshold.

1. Dynamic adjustment based on feedback of detection. Under the condition $V(q_{ij})<\delta_{aij}$, or $\text{distr}(r_{sa_i},r_s)>A_{ai}$, if it is not a business anomaly determined by experience, the threshold should be adjusted to $\delta_{aij}=V(q_{ij})$, or $A_{ai}=\text{distr}(r_{sa_i},r_s)$.

2. Dynamic adjustment based on business cycle. In business data, set the time point of a concerned business cycle is ExpireTimeStamp. For the business data satisfying $\text{TimeStamp}>\text{ExpireTimeStamp}$, formula (3), (4) and (5) are applied to recalculate the threshold. The threshold is periodically updated to readjust.

After the dynamic threshold of the anomaly detection is established, the service execution process in the collaborative task is detected online according to the detection threshold, and the detection algorithm is as shown in Algorithm 1.

---

**Algorithm 1: Business anomaly detection based on QoS benchmark (QoS_B_AD)**

| Input parameters:  | $r_{sa_i}$, the resource service used by a business activity  |
|--------------------|-------------------------------------------------------------|
|                    | $RS_i$, the resource service category to which $r_{sa_i}$ belongs, |
|                    | Corresponding QoS benchmark                                 |
| Output parameter:  | flag, whether the business process is normal,              |
|                    | data, anomaly data                                        |
| 1: $\delta_{aij} = 0$; $\delta_{aij}' = 0$; $A_{ai} = 0$; flag = true; data = null; |
| 2: for each rs      |                                                             |
| 3: if($r_s \in \text{instanceof}(RS_i)$) then              |
| 4: for each $V(q_{ij}) \in rs$                             |
| 5: if($\delta_{aij}>V(q_{ij})$) then $\delta_{aij}=V(q_{ij})$; // Update $\delta_{aij}$ |
| 6: else($\delta_{aij} \neq 0$ & $\delta_{aij}'<V(q_{ij})$) then $\delta_{aij}'=V(q_{ij})$; // Update $\delta_{aij}'$ |
| 7: if($A_{ai}<\text{distr}(r_{sa_i},r_s)$) then $A_{ai}=\text{distr}(r_{sa_i},r_s)$; // Update $A_{ai}$ |
| 8: end for                                                  |

---
9:  end if
10:  end for
11: for each \( V(q_{ij}) \in rs_{ai} \);  
12:  if \( V(q_{ij}) < \delta_{aij} \) \( || \) \( V(q_{ij}) > \delta'_{aij} \) then flag = false; data = \( V(q_{ij}) \); // Return QoS value
13:  if \( \text{distr}(rs,rs_{rb}) > \Delta_{ai} \) then flag = false; data = \( \text{distr}(rs,rs_{rb}) \); // Return anomaly data similar to the resource service chain QoS benchmark
14: end for
15: return flag, data

The time complexity of the algorithm is \( O(L*\text{len}(rs)) \). Where \( L \) is the number of business data, and \( \text{len}(rs) \) is the number of QoS indicators for each resource service.

4. Experiment and evaluation

4.1 Experimental verification

A collaborative design and production process of the electrical appliance is taken as an example, as shown in figure 1. There are 12 business activities altogether. The resource services that the activities use are listed in table 1.

![Figure 1. Electrical collaborative design and production process.](image)

From Figure 1 and Table 1, According to the sequence of activities, there are three RSCs, including: \(<rs_{1}, rs_{2}, rs_{3}, \{rs_{6}, rs_{7}, rs_{8}\}, rs_{10}, rs_{12}, \{rs_{13}, rs_{14}\}, rs_{15}>, <rs_{1}, rs_{2}, rs_{4}, rs_{10}, rs_{12}, \{rs_{13}, rs_{14}\}, rs_{15}>, <rs_{1}, rs_{2}, rs_{3}, rs_{6},\{rs_{11}, rs_{8}\}, rs_{12}, \{rs_{13}, rs_{14}\}, rs_{15}>\). The first RSC is chosen for analysis.

| Activities | Resource services          | Activities | Resource services          |
|------------|---------------------------|------------|---------------------------|
| \( a_1 \)  | \( RS_1: \) Market research report | \( a_7 \)  | \( RS_{16}: \) Configuration scheme |
| \( a_2 \)  | \( RS_2: \) General design scheme | \( a_8 \)  | \( RS_{16}: \) Test Engineer |
| \( a_3 \)  | \( RS_3: \) Hardware designer | \( a_9 \)  | \( RS_6: \) Machining equipment |
| \( a_4 \)  | \( RS_4: \) Software designer | \( a_{10} \) | \( RS_{12}: \) Processing staff |
| \( a_5 \)  | \( RS_5: \) Mechanical structural designer | \( a_{11} \) | \( RS_{15}: \) Assembly staff |
| \( a_6 \)  | \( RS_6: \) Parts            | \( a_{12} \) | \( RS_{14}: \) Inspect equipment |
| \( a_7 \)  | \( RS_7: \) Hardware processing equipment |                      | \( RS_{15}: \) Inspector |
| \( a_8 \)  | \( RS_8: \) Processing staff |                      | \( RS_{15}: \) Freight company |

Taking activity \( a_1 \) as example, the steps of calculation are listed as follows.
Step 1. According to formula (1) and (2), the similarities between RSC instances are calculated. The QoS benchmarks of the RSCs are resolved by k-means clustering algorithm. The QoS benchmarks of RS1 are Cost=0.259; Time=0.760; Availability=0.968 and Qualification=0.918.

Step 2. Dynamic adjustment. In the business data DB, a fragment of business data, as a business cycle, is selected. \( \delta_{wi} \), \( \delta_{wij} \) are calculated according to the formula (3) and (4) respectively. The result is: \( \delta_{wi}(\text{Cost})=0.18 \), \( \delta_{wij}(\text{Cost})=0.46 \); \( \delta_{wi}(\text{Time})=0.64 \), \( \delta_{wij}(\text{Time})=0.84 \); \( \delta_{wi}(\text{Availability})=0.83 \), \( \delta_{wij}(\text{Availability})=0.77 \); \( \delta_{wi}(\text{Qualification})=0.95 \) and \( \delta_{wij}(\text{Qualification})=0.95 \).

Step 3. According to formula (5), \( A_{ai} \) can be calculated, \( A_{ai} = 0.276 \). The business data to be detected are listed in table 2 according to the threshold.

| Rsc_id | Business activity id | Resource service type | Cost | Time | Availability | Qualification | TimeStamp         |
|--------|----------------------|-----------------------|------|------|--------------|---------------|-------------------|
| 001    | 1                    | RS1                   | 0.25 | 0.82 | 0.920        | 0.879         | 1510243200        |
| 002    | 1                    | RS1                   | 0.59 | 0.48 | 0.891        | 0.737         | 1510329600        |
| 003    | 1                    | RS1                   | 0.30 | 0.79 | 0.864        | 0.856         | 1510416000        |
| 004    | 1                    | RS1                   | 0.45 | 0.65 | 0.840        | 0.782         | 1510502400        |

Because QoS indicator Cost is 0.59, more than \( \delta_{wij}=0.46 \), and the similarity between the QoS of resource service with \( \text{Rsc}_\text{id} = '004' \) and the QoS benchmark is 0.288, more than \( A_{ai}=0.276 \), there is business anomaly in activity \( a_i \) at timestamp ‘1510329600’ and ‘1510502400’.

4.2 Results and Discussions
We make comparisons with QoS indicator detection method and fixed threshold method, as shown in figure 2.

![Figure 2. Comparing accuracy with other methods.](image)

It can be seen from the results of figure 2 that the QoS indicator detection method only detects the resource service used by individual business activity. Therefore the accuracy of the detection is always at a low level. Although a higher initial detection accuracy can be seen in the fixed threshold method, it generally follows a downward trend. This is because even if the threshold is set from two dimensions of the QoS indicator and the QoS benchmark of the RSC, threshold adjustment cannot be performed in the method. In QoSAD, business anomaly can be detected not only in local individual business activity but also in global business process aspects. It can maintain a stable and high detection accuracy in collaborative task.

5. Conclusions
To detect business anomaly in collaborative tasks, a business anomaly detection method based on the resource service chain QoS benchmark is proposed. By clustering QoS values into benchmark, a dynamic adjustment strategy to threshold is used. The simulation experiment demonstrates that QoSAD has higher detection accuracy than other two methods.
Acknowledgments
This work was supported in part by a grant from JZ160409 of the Natural Foundation Key Program for Young Scholars in the Universities of Fujian Province, 2018C110R of the Quanzhou Science and Technology Plan, and 2019H01010129 of the Fujian Province Science and Technology Plan.

References
[1] Bezerra, F., Wainer, J. (2013) Algorithms for anomaly detection of traces in logs of process aware information systems. Information Systems, 38(1): 33-44.
[2] Mi H, Wang H, Yin G, et al. (2011) Performance problems online detection in cloud computing systems via analyzing request execution paths. In: 2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W). Hong Kong. 135-139.
[3] Appel S., Frischbier S., Freudenreich T., et al. (2013) Event stream processing units in business processes. In: Weber, Barbara. (Eds.), Business Process Management. Springer, Berlin, Heidelberg. 187-202.
[4] Bao L., Li Q., Lu P., et al. (2018) Execution Anomaly Detection in Large-scale Systems through Console Log Analysis. Journal of Systems and Software, 143: 172-186.
[5] Seth, S., Singh, N. (2017) Dynamic Threshold-Based Dynamic Resource Allocation Using Multiple VM Migration for Cloud Computing Systems. In: International Conference on Information, Communication and Computing Technology. Singapore. 106-116
[6] Ali, M. Q., Al-Shaer, E., Khan, H., & Khayam, S.A. (2013) Automated anomaly detector adaptation using adaptive threshold tuning. ACM Transactions on Information and System Security, 15(4):1-30.
[7] Dani, M. C., Jollois, F. X., Nadif, M., Freixo, C. (2015) Adaptive threshold for anomaly detection using time series segmentation. In: International Conference on Neural Information Processing. Istanbul. 82-89.
[8] Li, H., Liang, M., He, T. (2017) Optimizing the composition of a resource service chain with inter-organizational collaboration. IEEE Transactions on Industrial Informatics, 13(3): 1152-1161.
[9] Li, H., He, T. (2018) Selecting Key Feature Sequence of Resource Services in Industrial Internet of Things. IEEE Access, 6(1): 72152-72162.