IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 23, 2019, accepted November 10, 2019, date of publication November 13, 2019, date of current version November 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953283

Detecting a Business Anomaly Based on QoS Benchmarks of Resource-service Chains for Collaborative Tasks in the IoT

HAIBO LI[1,2], JUNCHENG TONG[1], SHAOYUAN WENG[1], XINMIN DONG[1], AND TING HE[1]
[1]College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China
[2]Xiamen Engineering Research Center of Enterprise Interoperability and Business Intelligence, Huaqiao University, Xiamen 361021, China

Corresponding author: Haibo Li (lihaibo@hqu.edu.cn)

This work was supported in part by the National Key Research and Development Program of China, under Grant 2018YFB1402500, in part by the Fujian Province Science and Technology Plan under Grant 2019H0017, in part by the Quanzhou Science and Technology Plan under Grant 2018C110R, and in part by the Natural Foundation Key Program for Young Scholars in the Universities of Fujian Province under Grant JZ160409.

ABSTRACT In the Internet of Things (IoT), the online performance of many online services is determined by their distribution resources, which are connected to many different devices. The expected performance of a resource service primarily depends on the optimal use of the service in satisfying end-to-end quality requirements to support its successful execution. Therefore, the performance of a resource service is dynamic and should be discovered as a benchmark to detect a performance anomaly online. A performance anomaly is referred to as a business anomaly because it depends on its usage. The performance is measured by the quality of service (QoS) that is possessed by a resource service. In this paper, an approach based on the resource service QoS is proposed to detect a business anomaly via mining business process data in collaborative tasks in the IoT. First, a resource-service chain (RSC) is considered to be an analysis object because resource services are employed as a ‘‘service flow’’ by a business process. The similarity between any two RSCs is measured according to the QoS indicator values of resource services. Based on the similarity, a clustering algorithm is presented to resolve clustering centers that are considered to be QoS benchmarks. Second, according to the QoS benchmarks of RSCs, the thresholds of QoS indicators of a business anomaly are determined. Third, an algorithm is presented to detect anomalies of the business process. Finally, the proposed approach is illustrated by a simulation experiment. The experimental results show that the approach can be used to effectively detect a business anomaly online.

INDEX TERMS Internet of Things, collaborative task, resource service chain, benchmark of QoS, business anomaly.

I. INTRODUCTION

The Internet of Things (IoT) provides a new way for a wide range of manufacturing resources to optimize management and dynamic scheduling [1]. Collaborative manufacturing is a traditional application domain of the IoT. By merging manufacturing and service, decentralized manufacturing resources are integrated to ensure that business processes, in collaboration with different enterprises, are more competitive than ever [2]. A manufacturing resource, as an object that is directly applied to enterprise collaboration, provides service to a business process, which renders it is an important approach to interoperations among different enterprises. Supported by new information technologies, such as cloud computing, the IoT, CPSs, and big data, resource services that are extensively distributed in enterprises are integrated and scheduled via a workflow management system on a collaboration platform [3]. For example, to fulfill an order for electrical appliance manufacturing, a business process is implemented by a product design company, a software company, two manufacturing factories, and a final assembly factory, as shown in Fig. 1.

For a business process in a certain field, the service level and service capacity of a resource will be gradually

The associate editor coordinating the review of this manuscript and approving it for publication was Chun-Wei Tsai.

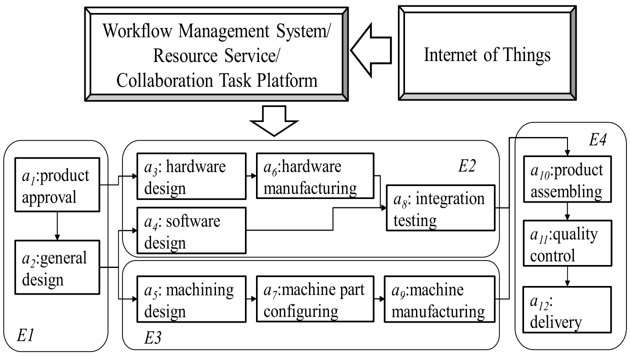

FIGURE 1. Process of electrical appliance collaborative design and manufacturing.

stabilizing. However, from a long-term perspective, the stabilization is not absolute because it will be volatile within a complex and changing market environment. We suggest that the business trend is changed. If the service capacity is greatly changed in a short time due to by various uncertainties internal or external to enterprises, a business anomaly occurs. For example, business stagnation or failure will occur due to an increased service cost, delayed service time or reduced yield rate. The key importance of detecting a business anomaly is attributed to the fact that bottlenecks of resource services occur over time and are eliminated by readjusting and optimizing the resource services. Thus, enterprises can collaborate more efficiently than ever.

The service capacity of a resource can be measured by the QoS (Quality of Service) [4]. For a whole business process, resource services are utilized as "flow". If some resource services are employed in a certain sequence, they form a chain of resource services—the Resource-Service Chain (RSC) [5]. Therefore, the service capacity of an RSC should be measured by a sequence of resource service QoSs instead of an individual resource service QoS. Consequently, evaluating the QoS of an RSC can benefit by grasping the service capacity of an entire business process. To detect a business anomaly, a typical business should be detected and identified, that is, benchmarks for an RSC should be established. However, accurately defining a dynamic real-time QoS benchmark of a resource service is difficult because it depends on too many changing factors, such as the type of business activity that a resource service provides and the dynamic market environment. Therefore, from this perspective, at least one benchmark exists for a given RSC. Currently, mining the QoS benchmark of an RSC by analyzing business data and then detecting a business anomaly should be further investigated. We refer to this problem as the DBARSC (Detecting Business Anomaly of Resource Service Chain).

In this paper, the detecting method that we propose is referred to as the DBAQoSB (Detecting Business Anomaly method of RSC based on QoS Benchmark). The DBAQoSB consists of a clustering algorithm to resolve the QoS benchmark of an RSC and a detecting algorithm to identify a business anomaly. Supported by the DBAQoSB, business decision makers in enterprises can readjust resource services as soon as possible to improve the resource utilization.

The contributions of the proposed method in this paper are as follows.

(1) A global perspective is proposed to detect business anomaly for collaborative tasks in the IoT. In collaborative environment, anomaly of an inter-organizational business process is difficult to be detected as resource services are used by each organization independently. In our method, an inter-organizational resource service sequence is considered to build benchmarks for anomaly detection. Also, the precision of detection is improved through global similarity between executing business data and QoS benchmarks of RSCs from historical data.

(2) The proposed method can be widely used to detect anomalies for various resource services in different fields, not limiting to some specific resource services, such as net resource services and storage resource services, etc.

The remainder of the paper is organized as follows: In Section 2, we present a survey of related studies and discuss different related approaches. In Section 3, we describe the characteristics of the DBARSC and its formulation. Section 4 presents an algorithm for clustering the QoS benchmark of an RSC. Section 5 presents another algorithm for detecting a business anomaly. The evaluation criteria, the design of the experiments and the results are presented in Section 6. In Section 7, we present a summary and discuss future research.

II. RELATED WORK

IoT technologies can be applied to various application domains because the uncertainty of the physical world can be ideally controlled using cyber technology [6]. The existing research interests include intelligent manufacturing [7], [8], energy management [9], traffic volume management, and healthcare systems [10]. The IoT, as a network infrastructure, enables the interoperability of these devices [11]. In a collaborative task system, efforts are made to overcome drawbacks that are caused by cross-domain communications and resource management [12], because system models are located far from the point of data generation.

In a collaborative task, more than one enterprise cooperates with each other to accomplish a common business process. A sequence of resource services, characterizing a temporal relation [2]–[5], provides a global perspective for measuring the attributes of resource services [13], [14]. A variety of methods for the evaluation of a resource service using QoS have recently been proposed. Most of the methods establish QoS evaluating models of resource or service for resource management [15] or service composition [16]. Others present mathematic models for the optimal composition of resource services [17] or optimize multiple metrics of QoS to satisfy the constraint of resource services [18]. Based on the QoS model, a two-sided matching model that is based on QoS is proposed to avoid dishonest evaluation [19]. Even though these methods have been shown to be effective, they are not used for a sequence of resource services.

A resource-service chain (RSC) is a sequence of resource services that is used by activities with the execution of a business process. Generally, an RSC can be obtained from a workflow model [3]–[5]. An RSC generally serves the purpose of selecting or compositing resources for business processes. In [3] and [5], composition methods of a resource-service chain, which are based on the use frequency of an RSC but not the QoS, are presented. In [2], features of resource services in an RSC, as a QoS indicator, are selected to identify key RSCs. In [20], a similar QoS evaluation model is used to resolve the optimal compositions of resource services within a business process. In [16], the service features in various service domains are defined, and their influences on the QoS of service composition are described.

A QoS benchmark is usually employed to describe the criteria of service, including the resource service. The optimal solution for the execution time or throughput QoS criteria can be determined in polynomial time; however, optimality is not guaranteed in polynomial time for the QoS criteria [21]. In [22], based on the QoS, an approach is presented to enable rapid, optimized deployment of software in a cloud environment by substantially reducing the number of required benchmarks. In [23], an application for monitoring and benchmarking individual application components is developed. In [24], a QoS-aware method is presented to monitor the QoS and estimate the cluster capacity that is required to satisfy the QoS benchmarks. In addition, clustering methods [25]–[27] can be used to mine QoS benchmarks. In [28], the QoS benchmark is described in APIs in a QoS-oriented modeling framework for automated cloud resource allocation and optimization.

Anomaly detection is an important problem that has been researched within diverse research areas and application domains. Anomaly detection involves identifying items, events, or observations that do not conform to an expected pattern or a model of normal behavior [29]. The application fields of anomaly detection include network intrusion detection [30], financial fraud detection, medical sensor detection, and fault detection in industrial systems [31]. Three categories of research exist: methods based on threshold features, methods based on statistical model features and methods based on performance features. In [32], a wavelet-based method is presented to detect an anomaly using discrete wavelet transforms to decompose real-time workload traces into multiple curves with different frequencies. Statistical analysis is also applied to decomposed traces. Currently, most studies focus on the performance anomaly. In [33], a dynamic threshold-based dynamic resource allocation method is presented to optimize the resource utilization and time. In [34], a virtualized platform is developed to predict the performance anomaly. In [35], an underlying temporal property of the stream via adaptive learning is estimated, and robust control charts are applied to recognize deviations. In [36], a hybrid deep-learning-based anomaly detection scheme for

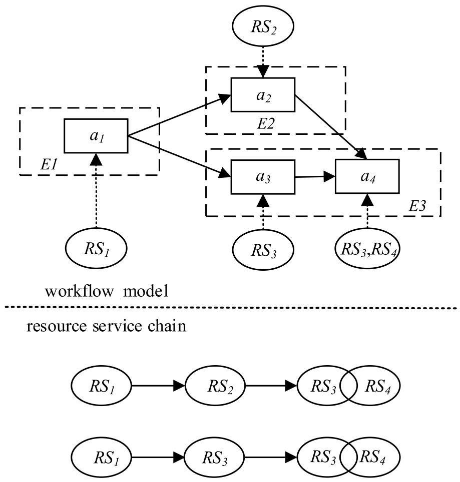

FIGURE 2. RSC in a collaborative task.

suspicious flow detection in the context of social multimedia is proposed to satisfy strict QoS requirements. For business process, detecting the anomaly of a data model can be solved by a data-process graph model [37]. In [8], dynamic QoS anomaly prediction to predict network anomalies, such as latency and backlog, is discussed; however, it is not applied to predict business anomalies. Though in these methods, the QoSs of single resource service are usually used to detect various anomalies, multiple QoS-based sequence of resource services are rarely not considered. The QoS benchmarks of RSCs could be built as a criterion for anomaly detection.

As previously described, current research on the detection of anomalies are primarily based on QoS evaluation models. However, methods for the sequence of resource services that consider a self-adapting threshold are rare.

III. PROBLEM FORMULATION

Generally, disperse resource services are scheduled via workflow technology in a collaborative task system. In addition, the order of business activities and resource services required by business activities are defined in a workflow model, as shown in Fig. 1. The workflow can be defined as follows.

Definition 1 (Workflow): The workflow is defined as a 4-tuple, $Wf = (id, Activity, \prec, R)$, where id is a unique identifier of the workflow; $Activity$ is a set of business activities that are executed; $\prec$ is a specification of an order between activities for which any activities $A_i, A_j \in Activity$, $1 \leq i, j \leq n$, n is the number of business activities; $A_i \prec A_j$ indicates that A_i is executed before A_j, i.e., A_i precedes A_j in a workflow instance; and R is a set of resource services.

Definition 2 (RSC): For any workflow Wf, a resource service chain is denoted as $RSC = \{<RS_1, RS_2, \ldots, RS_n> | n \leq |Activity|\}$, where $RS_1, RS_2, \ldots, RS_n$ are the sets of resource services used by activities.

An RSC is a sequence of resource services. Given a workflow model, as shown in Fig. 2, two possible RSCs, which are denoted as $<RS_1, RS_2, \{RS_3, RS_4\}>$ and $<RS_1, RS_3, \{RS_3, RS_4\}>$, exist.

Definition 3 (QoS Benchmark of RSC): The QoS Benchmark of RSC is denoted as $QoSB = \{<Index_1, Index_2, \ldots, Index_n> | n \leq |Activity|\}$, where $Index_n$ is the QoS indicator values set of RS_n in an RSC, which represents the QoS benchmark of the resource service RS_n.

The QoS indicator value is usually referred to as a QoS value. The QoS benchmark of an RSC is a set of QoS values, which represents the service capacity of a resource service sequence. We are interested in determining the QoS benchmark of an RSC via analyzing business data. The resource services, QoS values of resource services and their timestamps must be recorded in business data. Business data can be defined as follows:

***Definition** 4 Business Data*. Business data is denoted as $BD = \{aid, rsid, RS, Index, TimeStamp\}$, where aid is a unique identifier of business activity, $rsid$ is a unique identifier of resource service instance, RS is a set of resource service instances, $Index = \{V(q_1), V(q_2), \ldots, V(q_n)\}$,$V(q_i)$ is a value set of QoS indicator q_i, and $TimeStamp$ is a set of times when a resource service RS is used by activities.

***Definition** 5 Business Anomaly (BA)*. Business anomaly is denoted as $BA = \{aid, rsid, R, Threshold, AbnormalIndex\}$; aid is a unique identifier of business activity; $rsid$ is a unique identifier of resource service instance; R is a set of resource service categories; $Threshold = \{threshold_i | i = 1, \ldots, n, n = |R|\}$; $Threshold_i = \{\delta_{ai1}, \ldots, \delta_{aik}, \delta'_{ai1}, \ldots, \delta'_{aik}, \Delta_{ai}\}$, k is the number of QoS indicators for $rs_i\}$, where δ_{aij} and δ'_{aij} are the threshold lower bound and threshold upper bound, respectively, of the *j-th* QoS indicator q_j; Δ_{ai} is the threshold of the QoS benchmark; and $AbnormalIndex$ is a set of abnormal QoS indicators.

For any $index_i$ in $AbnormalIndex$, let δ_{aik} and δ'_{aik} be the threshold of $index_i$, then $\delta_{aik} > index_i$ and $\delta'_{aik} < index_i$.

For the workflow Wf and a set of resource service R, we want to resolve the QoS benchmark of an RSC $QoSB$ by analyzing business data DB and then resolve the set of business anomalies $AbnormalIndex$.

IV. ESTABLISHING QOS BENCHMARK OF RSC

A. EVALUATING QOS INDICATOR OF RESOURCE SERVICE

In contrast to a general web service, resource services that use collaborative tasks vary in different fields. Even in the business process in the same field, the categories of resource services differ. Therefore, QoS indicators differ. For example, in the manufacturing field, depending on the different roles of resource services, resource services can be divided into eight categories: equipment, material, human resource, field, computing, knowledge, software and other resource services [5]. In this classification, QoS indicators of equipment focus on the utilization rate, availability and working hour, whereas QoS indicators of human resource focus on the service cost, service qualification, and service reputation.

Both quantitative QoS indicators and qualitative QoS indicators should be considered to evaluate a resource service. Qualitative QoS indicators such as qualification, reputation and security need to be measured by grading and quantifying. Because QoS indicators have different units and ranges of values, their direct use will have a considerable influence on the calculation results. Therefore, the QoS values should be scaled to eliminate these influences. Assume that q_{ij} is the value of the *j-th* QoS indicator for the resource service rs_i and $V(q_{ij})$ is the QoS value after being scaled. The QoS value can be scaled in the following two categories.

(1) Positive indicator. A positive indicator with a higher value is better. Examples of positive indicators include security, technology level, and reputation. The following formula is used to measure the scaled values of the positive indicator.

$$V(q_{ij}) = \frac{q_{ij}}{\max\limits_{j}(q_{ij})} \tag{1}$$

(2) Negative indicator. A negative indicator with a lower value is better. Examples of negative indicators include service time and service cost. For easy calculation, the following formula is used to measure the scaled values of the negative indicator. After being scaled, a higher QoS value is better.

$$V(q_{ij}) = \frac{\min\limits_{j}(q_{ij})}{q_{ij}} \tag{2}$$

B. SIMILARITY COMPARISON OF RSCs

A comparison of the similarity between QoS indicators in the same category is meaningful because QoS indicators from different categories of resource services have different semantics. Assume that k is the number of QoS indicators for a category of resource services; and $V(q_{aj})$ and $V(q_{bj})$ are the *j-th* QoS values of the resource service rs_a and rs_b, respectively. The following formula is used to measure the distance between the resource services rs_a and rs_b.

$$disr(rs_a, rs_b) = \sqrt{\sum_{j=1}^{k} [V(q_{aj}) - V(q_{bj})]^2} \tag{3}$$

Based on (3), the distance between two RSCs can be considered as a merged distances each of which is determined via a pair of resource services in two RSCs. In addition to the pair of the resource services at the corresponding position of the two RSCs, the resource services at different positions of two RSCs should also be measured if they are used by different business activities, such as resource service RS_3 in Fig. 2. Therefore, to calculate the distance between RSCs, both local and global distance should be reflected. The following formula is given as follows to calculate the distance between two RSCs.

$$disrsc(rsc_n, rsc_m) = \max_{j=1}^{l} disr(rs_{nj}, rs_{mj}) * \sum_{j=1}^{l} disr(rs_{nj}, rs_{mj}) \tag{4}$$

Algorithm 1 CluQBRSC // Clustering QoS Benchmark for RSC

Input: $RSC = \{rsc_1, \ldots, rsc_n\}$, a set of RSCs; k, the number of cluster

Output: $O = \{o_1, \ldots, o_k\}$, k clustering centers; $C = \{c_1, \ldots, c_k\}$, k clusters

1: select $O = \{o_1, \ldots, o_k\}$ randomly from RSC as the initial cluster centers;

2: while $E = \sum E_i | i = (1, \ldots, k)$ //no longer change
  for each rsc_i in RSC do Insert rsc_i to c_j where $disrsc_j = \{\min[disrsc(rsc_i, o_j)] | i = (1, \ldots, k)\}$; //assign other RSCs to the similar cluster.

3: for each c_i in c do $E_i = (\sum disrsc\ (o_i, rsc_j) | rsc_j \in c_i)$; //calculate the distance difference between center and sum of other RSCs in current cluster.

4: Select $o_{temp} (o_{temp} \neq o_i)$ randomly from c_i;

5: $E_{temp} = (\sum dist(o_{temp}, rsc_j) | rsc_j \in c_i)$;

6: if($E_{temp} <$ E$_i$) then $o_i = o_{temp}$; //update the cluster center

7: end while

8: return O, C;

In (4), rsc_n and rsc_m are two RSCs, l is the length of the RSCs, rs_{nj} and rs_{mj} are two resource services in the two RSCs. According to (3) and (4), rsc_n and rsc_m are more similar if both the sum and the maximum value of the distance between two resource services at corresponding position of the two RSCs are smaller.

C. MINNING ALGORITHM OF QOS BENCHMARK FOR RSC

Similar QoS values of a resource service manifest a similar service capacity. Even for the same resource service, its QoS value may differ among different business activities or processes. Therefore, a clustering algorithm would be suitable for mining the QoS benchmark. The algorithm CluQBRSC, which is based on k-means, is presented to obtain all QoS benchmarks for RSCs.

In algorithm 1, initial k cluster centers are selected. Assigning $RSC = \{rsc_1, \ldots, rsc_n\}$ to the most similar clusters, repeat and update until all cluster centers do not change. A major factor that affects algorithm performance is the number of elements in the set RSC, which is denoted as $|RSC|$, and the number of clusters k. The algorithmic complexity is O ($|RSC|$).

The final result of algorithm 1 is k cluster centers. Each cluster center represents a QoS benchmark of RSCs. These QoS benchmarks provide support for selecting a resource service for various requirements of a business process.

V. DETECTING BUSINESS ANOMALY

A. DETERMINING THRESHOLD OF BUSINESS ANOMALY DETECTION

During a normal business process, the QoS values of the selected resource services should fall within a reasonable range. If one or more indicators deviate from the range, some activities are abnormal and may influence the following activities, such as far-exceeding the service time and service cost. In addition to the resource service, the QoS values of RSCs should also fall within a reasonable range. Otherwise, an abnormal business process may influence the collaboration with other business processes. Therefore, QoS values of resource services and distances from the corresponding QoS benchmark should be determined.

(1) The threshold of the QoS indicator. If activity a selects resource service rs_i, and the j-th QoS indicator of rs_i has the lower bound δ_{ij} and the upper bound δ'_{ij}, where rs_i is the instance of resource service RS_i, let $V(q_{ij})$ be the QoS value after scaling. The lower bound and upper bound can be measured by the following formula.

$$\delta_{ij} = \min_i(V(q_{ij})) \tag{5}$$

$$\delta'_{ij} = \max_i(V(q_{ij})) \tag{6}$$

(2) The threshold of distance with the QoS benchmark. If activity a selects resource service rs_i, let $Index_{ij}$ be the benchmark of the j-th QoS indicator, and let Δ_{ij} be the threshold of distance with the QoS benchmark $Index_{ij}$. Δ_{ij} can be measured by the following formula.

$$\Delta_{ij} = \max_i(|V(q_{ij}) - Index_{ij}|) \tag{7}$$

B. DYNAMIC CALIBRATION OF THRESHOLD

The requirement would be dynamically adapted with a changing market. For example, an increased service cost due to rising prices of raw materials, and a shortened service time caused by an improved technology. The threshold for detecting a business anomaly should be dynamically calibrated to ensure that the accuracy of detection can be guaranteed. The dynamic calibration of threshold should follow two methods.

(1) The dynamic calibration based on the feedback of detection. If the QoS values of the business anomaly cannot be accepted after manual confirmation, the threshold of the anomaly should be calibrated to the correct threshold. For example, the QoS value $V(q_{ij})$ is less than δ_{ij}, or the difference between $V(q_{ij})$ and $Index_{ij}$ is more than Δ_{ij}. If this value is not a business anomaly after manual confirmation, the threshold of the QoS anomaly should be calibrated as $\delta_{ij} = V(q_{ij})$ or $\Delta_{ij} = V(q_{ij})$ - $Index_{ij}$.

(2) The dynamic calibration based on available time span. In business data, all resource services are recorded by timestamps. However, larger historical data does not mean better detection. Only the data in recent period achieves a best-value for current business. Therefore, the reasonable time span should be established according to a business scenario. The time-expired data will gradually lose value for the current business and even disturb the accuracy of the algorithm, so that it should be discarded. The threshold should be recalculated according to formulas (5), (6) and (7) and periodically updated to ensure that it can be dynamically calibrated.

Algorithm 2 DBARQoSB (Detecting Business Anomaly of RSC Based on QoS Benchmark)

Input: BD, business data set
RS, a category of resource service, where $RS \in R$
$V(q_{ij})$, j-th QoS value of resource service rs_i, rs_i is an instance of RS
$Index_{ij}$, benchmark of $V(q_j)$
Output: q, QoS value of resource service there is business anomaly, if null, no business anomaly

$flag$ = true; q = null;
for each $rs_i \in BD$
  if($rs_i \in instanceof(RS)$)
  for each $V(q_{ij})$ of rs_i
    if($\delta_{ij} > V(q_{ij})$) then $\delta_{ij} = V(q_{ij})$; //update δ_{ij}
    else if ($\delta'_{ij} \neq 0$ &&$\delta'_{ij} < V(q_{ij})$) then $\delta'_{ij} = V(q_{ij})$; // update δ'_{ij}
      end if
      if($\Delta_{ij} < V(q_{ij})$ - $Index_{ij}$) then $\Delta_{ij} = V(q_{ij})$ - $Index_{ij}$; //update Δ_{ij}
      end for
  end if
end for
  for each $V(q_{ij})$ of rs_i
q = null;
if($V(q_{ij}) < \delta_{ij} || V(q_{ij}) > \delta'_{ij}$) then $q = V(q_{ij})$;
if($V(q_{ij})$ - $Index_{ij} > \Delta_{ij}$) then $q = V(q_{ij})$ - $Index_{ij}$;
end for
return q;

C. ALGORITHM OF DETECTING BUSINESS ANOMALY

After determining the threshold of business anomaly detection, algorithm 2 is presented to detect the business process online.

The main cost of algorithm 2 is to determine the threshold of detecting a business anomaly, depending on the data size and the number of QoS indications. The algorithmic complexity is O($|BD|$∗len(rs_i)), where $|BD|$ is the number of records indexed by activities, $V(rs)$ is the number of QoS indications in the resource service set rs, $rs_i \in instanceof(RS)$, and $RS \in R$.

VI. PERFORMANCE EVALUATION

A. EXPERIMENTAL SETUP

We consider the collaborative design and manufacturing processes of electrical apparatuses as a case to analyze our method, as shown in Fig 1. This case is a typical case in design and manufacturing of industrial products. Four business processes cooperate with each other: general designing, software design and hardware manufacturing, machining design and manufacturing, and product assembling. All resources that are required in the processes are listed in Table 1.

TABLE 1. Activities and resource services.

Business activities	Resource services
a_1:product approval	RS_1:market report
a_2: general design	RS_2:general design scheme
a_3: hardware design	RS_3: hardware designer
a_4: software design	RS_4: software designer
a_5: machining design	RS_5: machining designer
a_6: hardware manufacturing	RS_6:part RS_7: hardware manufacturing equipment RS_8: manufacturing worker
a_7: machine part configuring	RS_9:part supply scheme
a_8: integration testing	RS_{10}:testing engineer
a_9: machine manufacturing	RS_{11}: machine manufacturing equipment RS_8: manufacturing worker
a_{10}: product assembling	RS_{12}: assembling worker
a_{11}: quality control	RS_{13}: QC instrument RS_{14}:QC inspector
a_{12}: delivery	RS_{15}: transport supplier

According to the workflow model, as shown in Fig. 1, and different resource service categories of the activities, three possible RSCs exists at runtime: $chain_1 = <RS_1, RS_2, RS_3, \{RS_6, RS_7, RS_8\}, RS_{10}, RS_{12}, \{RS_{13}, RS_{14}\}, RS_{15}>$; $chain_2 = <RS_1, RS_2, RS_4, RS_{10}, RS_{12}, \{RS_{13}, RS_{14}\}, RS_{15}>$ and $chain_3 = <RS_1, RS_2, RS_5, RS_9, \{RS_{11}, RS_8\}, RS_{12}, \{RS_{13}, RS_{14}\}>, RS_{15}>$.

Due to limited space, RSC chain1 is selected for the analysis of the QoS benchmark. The simulation experiment has three steps: 1) preparing data; 2) resolving the QoS benchmark of the RSC; and 3) resolving the business anomaly.

Step 1) Preparing data.

In manufacturing, a standard data set of resource services that is open and accepted by most researchers does not exist. In current studies on resource service selection, simulation data are primarily employed. In our study, the production of high, medium and low end products is considered as an example to simulate the collaborative design and manufacturing processes. In our experiment, 500 RSCs, as a data set, are generated. Nine QoS indicators are used altogether, including service cost, service time, availability, service qualification, service reputation, yield rate, precision, technology level and safety. In these QoS indicators, service cost and service time are negative indicators. The minimum values for the same category of resource services are selected and then scaled by (2). For the other positive QoS indicators, the maximum values for the same category of resource service are selected and then scaled by (1). The segment of the preprocessed data set is shown in Table 2.

Step 2) Resolving QoS benchmark of the RSC.

According to algorithm 1, clustering centers are resolved. The number of clusters k is set to 3 because three product grades exist. Note that different values of k can be selected based on factors that influence the QoS of RSC. After clustering, three clustering centers are generated, as shown in Table 3.

From the results in Table 3, the QoSs vary in different clusters even for the same resource service due to the difference in business requirements. Therefore, the benchmarks of an RSC also differ.

TABLE 2. Segment of preprocessed data set.

RSC No.	A No.	CRS	SC	ST	Avb	SQ	SR
001	1	RS_1	0.786	0.284	0.591	0.557	-
001	2	RS_2	0.551	0.371	0.458	0.604	-
001	3	RS_3	0.857	0.321	0.606	0.490	0.570
001	6	RS_6	0.740	0.407	0.543	-	-
001	6	RS_7	0.811	0.394	0.474	-	-
001	6	RS_8	0.553	0.380	0.541	0.558	0.589
001	8	RS_{10}	0.838	0.267	0.604	0.424	0.458
001	10	RS_{12}	0.709	0.208	0.596	0.474	0.558
001	11	RS_{13}	0.767	0.323	0.459	-	-
001	11	RS_{14}	0.806	0.340	0.500	0.478	0.543
001	12	RS_{15}	0.800	0.217	0.589	-	-
002	1	RS_1	0.268	0.704	0.978	0.948	-
002	2	RS_2	0.199	0.722	0.965	0.938	-
002	3	RS_3	0.412	0.750	0.926	0.885	0.914
002	6	RS_6	0.247	0.688	0.875	-	-
002	6	RS_7	0.283	0.867	0.928	-	-
002	6	RS_8	0.263	0.842	0.927	0.916	0.937
002	8	RS_{10}	0.419	0.800	0.896	0.902	0.854
002	10	RS_{12}	0.252	0.625	0.883	0.917	0.904
002	11	RS_{13}	0.280	0.767	0.929	-	-
002	11	RS_{14}	0.260	0.850	0.904	0.880	0.876
002	12	RS_{15}	0.395	0.667	0.863	-	-

A=Activity, CRS=Category of Resource Service, SC=Service Cost, ST= Service Time, Avb=Availability, SQ= Service Qualification, SR=Service Reputation

TABLE 2. *(Continued.)* Segment of preprocessed data set.

RSC No.	A No.	CRS	YR	P	TL	S
001	1	RS_1	-	-	-	-
001	2	RS_2	-	-	0.340	-
001	3	RS_3	-	-	-	-
001	6	RS_6	0.755	-	-	-
001	6	RS_7	0.812	-	0.369	-
001	6	RS_8	0.680	-	-	-
001	8	RS_{10}	-	0.701	-	-
001	10	RS_{12}	-	-	-	-
001	11	RS_{13}	-	0.653	-	-
001	11	RS_{14}	-	0.740	-	-
001	12	RS_{15}	-	-	-	0.677
002	1	RS_1	-	-	-	-
002	2	RS_2	-	-	0.914	-
002	3	RS_3	-	-	-	-
002	6	RS_6	0.949	-	-	-
002	6	RS_7	0.927	-	0.813	-
002	6	RS_8	0.969	-	-	-
002	8	RS_{10}	-	0.948	-	-
002	10	RS_{12}	-	-	-	-
002	11	RS_{13}	-	0.969	-	-
002	11	RS_{14}	-	0.906	-	-
002	12	RS_{15}	-	-	-	0.949

A=Activity, CRS=Category of Resource Service, YR= Yield Rate, P= Precision, TL= Technology Level, S= Safety

Step 3) Resolving the business anomaly.

We consider the business activity a_1 as an example, which is in the business processes for high end products. The resource service RS_1 is used by the business activity a_1. The segment of preprocessed data is shown in Table 4.

TABLE 3. Clustering centers.

CC	RSC No.	A No.	CRS	SC	ST	Avb	SQ	SR
	034	1	RS_1	0.863	0.297	0.613	0.577	-
	034	2	RS_2	0.492	0.351	0.479	0.563	-
	034	3	RS_3	0.808	0.300	0.585	0.531	0.559
	034	6	RS_6	0.790	0.423	0.490	-	-
	034	6	RS_7	0.876	0.419	0.505	-	-
O1	034	6	RS_8	0.651	0.421	0.531	0.495	0.611
	034	8	RS_{10}	0.689	0.271	0.646	0.457	0.479
	034	10	RS_{12}	0.639	0.227	0.617	0.494	0.543
	034	11	RS_{13}	0.717	0.310	0.480	-	-
	034	11	RS_{14}	0.758	0.304	0.468	0.554	0.643
	034	12	RS_{15}	0.872	0.244	0.611	-	-
	478	1	RS_1	0.478	0.452	0.774	0.732	-
	478	2	RS_2	0.300	0.464	0.719	0.698	-
	478	3	RS_3	0.537	0.450	0.787	0.688	0.699
	478	6	RS_6	0.353	0.534	0.708	-	-
	478	6	RS_7	0.588	0.542	0.649	-	-
O2	478	6	RS_8	0.371	0.533	0.740	0.663	0.768
	478	8	RS_{10}	0.419	0.432	0.813	0.717	0.656
	478	10	RS_{12}	0.402	0.378	0.734	0.701	0.767
	478	11	RS_{13}	0.375	0.426	0.673	-	-
	478	11	RS_{14}	0.471	0.548	0.712	0.804	0.786
	478	12	RS_{15}	0.618	0.400	0.800	-	-
	165	1	RS_1	0.259	0.760	0.968	0.918	-
	165	2	RS_2	0.206	0.765	0.885	0.958	-
	165	3	RS_3	0.414	0.642	0.894	0.865	0.892
	165	6	RS_6	0.231	0.647	0.900	-	-
	165	6	RS_7	0.310	0.867	0.875	-	-
O3	165	6	RS_8	0.251	0.889	0.948	0.916	0.968
	165	8	RS_{10}	0.287	0.696	0.958	0.913	0.844
	165	10	RS_{12}	0.241	0.543	0.894	0.887	0.914
	165	11	RS_{13}	0.266	0.719	0.949	-	-
	165	11	RS_{14}	0.229	0.739	0.872	0.924	0.908
	165	12	RS_{15}	0.402	0.769	0.916	-	-

CC= Clustering Center, A=Activity, CRS=Category of Resource Service, SC=Service Cost, ST= Service Time, Avb=Availability, SQ= Service Qualification, SR=Service Reputation

According to the benchmark of RSC, the benchmarks of RS_1 include Service Cost (SC) = 0.259, Service Time (ST) = 0.760, Availability (Avb) = 0.968 and Service Qualification (SQ) = 0.918.

To eliminate the deviation for the thresholds caused by the time-expired data, for which the timestamp is greater than 1507601400. Note that different thresholds depend on different segment of business data. Manual confirmation is necessary. According to (5) and (6), δ_{aij} and δ'_{aij} are calculated, as shown in Table 5.

Based on Table 5 and according to (5), Δ_{ij} can be calculated. $\Delta_{ij} = 0.2755$. To detect a business anomaly, a segment of the data set is listed in Table 6. Two anomalies exist in resource services that are numbered 002 and 004, because the Service Cost (SC) is 0.586 > δ'_{ij} where $\delta'_{ij} = 0.458$, and the distance to the QoS benchmark is 0.2876 > Δ_i, where $\Delta_{ai} = 0.2755$. Therefore, business anomalies exist in business activity a_1 when it executes at timestamp 1510329600 and 1510502400.

TABLE 3. ***(Continued.)*Clustering centers.**

CC	RSC No.	A No.	CRS	YR	P	TL	S
O1	034	1	RS_1	-	-	-	-
	034	2	RS_2	-	-	0.383	-
	034	3	RS_3	-	-	-	-
	034	6	RS_6	0.735	-	-	-
	034	6	RS_7	0.792	-	0.427	-
	034	6	RS_8	0.639	-	-	-
	034	8	RS_{10}	-	0.680	-	-
	034	10	RS_{12}	-	-	-	-
	034	11	RS_{13}	-	0.602	-	-
	034	11	RS_{14}	-	0.718	-	-
	034	12	RS_{15}	-	-	-	0.697
O2	478	1	RS_1	-	-	-	-
	478	2	RS_2	-	-	0.660	-
	478	3	RS_3	-	-	-	-
	478	6	RS_6	0.878	-	-	-
	478	6	RS_7	0.854	-	0.635	-
	478	6	RS_8	0.784	-	-	-
	478	8	RS_{10}	-	0.835	-	-
	478	10	RS_{12}	-	-	-	-
	478	11	RS_{13}	-	0.786	-	-
	478	11	RS_{14}	-	0.813	-	-
	478	12	RS_{15}	-	-	-	0.818
O3	165	1	RS_1	-	-	-	-
	165	2	RS_2	-	-	0.904	-
	165	3	RS_3	-	-	-	-
	165	6	RS_6	0.959	-	-	-
	165	6	RS_7	0.917	-	0.823	-
	165	6	RS_8	0.948	-	-	-
	165	8	RS_{10}	-	0.928	-	-
	165	10	RS_{12}	-	-	-	-
	165	11	RS_{13}	-	0.980	-	-
	165	11	RS_{14}	-	0.917	-	-
	165	12	RS_{15}	-	-	-	0.960

CC= Clustering Center, A=Activity, CRS=Category of Resource Service, YR= Yield Rate, P= Precision, TL= Technology Level, S= Safety

TABLE 4. **A segment of preprocessed data for high end products.**

RSC No.	A No.	CRS	SC	ST	Avb	SQ	TS
001	1	RS_1	0.374	0.756	0.900	0.779	1507515000
002	1	RS_1	0.268	0.704	0.978	0.948	1507601400
003	1	RS_1	0.177	0.836	0.968	0.956	1507687800
004	1	RS_1	0.458	0.642	0.825	0.874	1507774200
005	1	RS_1	0.251	0.738	0.948	0.916	1507860600

RS=Resource Service, A=Activity, CRS=Category of Resource Service, SC=Service Cost, ST= Service Time, Avb=Availability, SQ= Service Qualification, TS=Timestamp

B. RESULTS AND DISCUSSIONS

The majority of similar research studies apply two categories of methods. The first method is based on QoS indicators of a single resource service. In this method, QoS indicators of a single resource service (named ISRS-QoS) are only used to detect a business anomaly, instead of the temporal relation between resource services in an RSC. The second method is based on a fixed threshold and is named FixedTh. In FixedTh, the threshold of the QoS benchmark cannot be dynamically calibrated.

TABLE 5. **Thresholds of QoS indicates for resource service RS_1.**

	SC	ST	Avb	SQ
δ_{aij}	0.177	0.177	0.825	0.774
δ'_{aij}	0.458	0.458	0.978	0.949

SC=Service Cost, ST= Service Time, Avb=Availability, SQ= Service Qualification

TABLE 6. **Segment of data set for detecting business anomalies.**

RSC No.	A No.	CRS	SC	ST	Avb	SQ	TS
001	1	RS_1	0.253	0.816	0.920	0.879	1510243200
002	1	RS_1	0.586	0.484	0.891	0.737	1510329600
003	1	RS_1	0.302	0.785	0.864	0.856	1510416000
004	1	RS_1	0.445	0.645	0.840	0.782	1510502400
005	1	RS_1	0.240	0.812	0.920	0.915	1510588800

RS=Resource Service, A=Activity, CRS=Category of Resource Service, SC=Service Cost, ST= Service Time, Avb=Availability, SQ= Service Qualification, TS=Timestamp

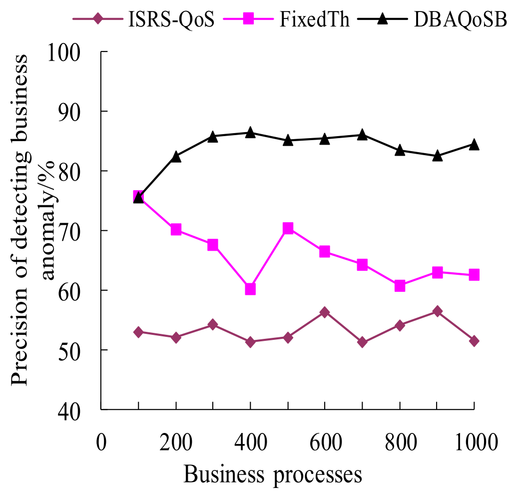

FIGURE 3. **Comparison of the three experiments.**

Considering the process of electrical appliance collaborative design and manufacturing as an example, as shown in Fig. 1, we make comparisons of precision for detecting business anomalies among ISRS-QoS, FixedTh and our method DBAQoSB, as shown in Fig. 3.

Because different business activities interoperate in a collaborative task, resource services are not employed in isolation. In ISRS-QoS, the resource services used by an individual activity are detected, which keeps the precision of detection at a relatively low level, as shown in Fig. 3. In FixedTh, a business anomaly is detected from the perspective of the QoS indicator. However, the precision is low without dynamic calibration of the threshold, because the business requirement would change in a long time window. Although the initial precision is high, a decreased trend is observed during the entire course. In addition to the advantage of FixedTh, our method DBAQoSB maintains high precision with dynamic calibration of the threshold. Therefore, our method is reliable.

C. EVALUATION USING ERP DATA

The collaborative business process of cement industry is taken as a case to analyze our method. The dataset is taken from a cement group company in Henan province of China, who undertakes the business of purchasing, inventory, manufacturing, logistics, and sale. In collaborative environment, the state of manufacturing equipment is collected via the IoT devices by the manufacturing factory, and the state of logistics vehicles is collected via the IoT devices by the logistics company. Analyzing the data from ERP, decision-makers are in a position to make more actionable decisions about their resources, their assets, their employees, their suppliers, and their customers [38].

The five business steps, including purchasing, inventory, manufacturing, logistics, and sale are taken as activities that compose a core business process. The resource services used by these activities include raw materials, purchasers and suppliers for purchasing activity, warehouses for inventory activity, equipment for manufacturing activity, vehicles for logistics activity and cement products for sale activity. In all possible RSCs, an RSC is selected that is followed in the order by raw materials, warehouses, equipment, vehicles and cement products. In the RSC, the QoSs of each resource service, including cost, amount, energy consumed, shipment accuracy, duration, qualification, and price are taken into consideration in part or whole.

In our method, the QoS values are used to discover the benchmarks of the RSC. Using the DBAQoSB method, a comparison is developed to check the precision of anomaly detection. To serve the purpose, three sizes of dataset, 2 months, 4 months and 6 months of historical data are respectively used to resolve the benchmarks. A detected object is also required to act as a business anomaly. For this purpose, a same RSC with aforementioned in the recent business process is selected. The precisions between the recent QoS values and the three benchmarks are compared for the five business activities, as shown in Fig. 4.

The overall trend shows that the benchmark resolved using the two months of data has the highest precision to detect anomaly of the recent business process. This is because that the precision has a great relation with the size of dataset for a given period of time. Specifically, it is suitable to detect business anomalies using the benchmark analyzed by the most recent data in the period.

An abnormal precision is shown in activity 4, i.e. the logistics, where 73.01% and 75.5% are for the benchmarks resolved through the 4 months of dataset and the 6 months of dataset respectively, as shown in Fig. 4. This is because that some QoS values were constantly changing either up or down during the recent 6 months period, as shown in Fig. 5. To resolve the benchmark, the size of 6 months of dataset is too bigger than that of the 4 months dataset. This shows that the result of the DBAQoSB method is consistent with practical business.

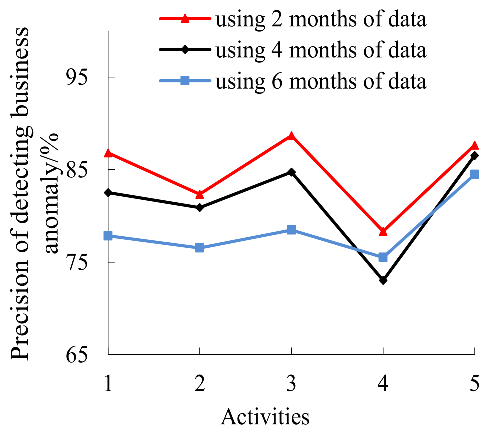

FIGURE 4. Comparison precision using three sizes of dataset.

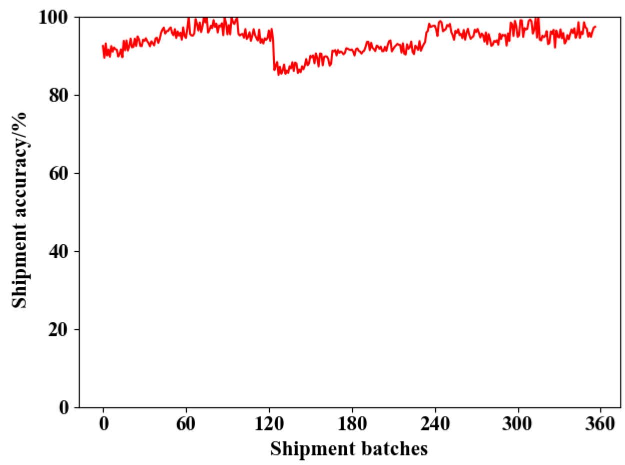

FIGURE 5. QoS values of the logistics activity.

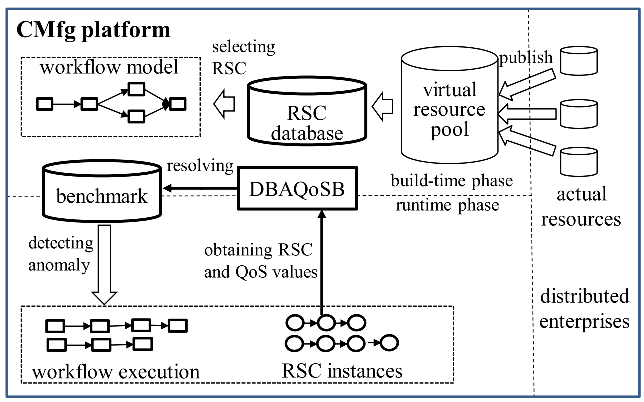

FIGURE 6. Application model of DBAQoSB on CMfg platform.

D. APPLICATION MODE

A collaborative manufacturing (CMfg) platform for small and medium-sized enterprises is developed by us in 2012.

On CMfg, dispersed and diverse manufacturing resources from different enterprises are accessed via IoT devices and integrated together to fulfill business through workflow system. In recent years, the DBAQoSB method is added into CMfg. The architecture of CMfg is shown in Fig. 6. The application mode of DBAQoSB is introduced as follows.

Before starting a workflow, manufacturing resource services are selected from virtual resource pool, in which all resource services are published by actual enterprises. According to workflow model and RSC instances generated by its continuous execution, DBAQoSB is applied to resolve the benchmarks. All these benchmarks are used to detect anomalies in each workflow. In addition, to keep a high precision, the benchmarks should be updated periodically by selecting a suitable size of history dataset and dynamically calibrating the thresholds of QoS indicators.

VII. CONCLUSION

A method for detecting a business anomaly based on the QoS benchmark DBAQoSB is presented for a collaborative task in the IoT environments in this paper. The method serves the purpose of establishing a QoS benchmark of an RSC, detecting and identifying the business anomalies of a business process. DBAQoSB is a practical method because the global information of similarity between the current resource service and QoS benchmarks of RSCs are considered. Therefore, DBAQoSB can guarantee that collaborative enterprises will identify business trends and business anomalies, which enables business decision makers to readjust resource services as soon as possible.

Future work will focus on the dependency among resource services in an RSC in a collaborative task. Different dependencies exist among resource services; they influence the usage of resource services. Especially in an RSC of a business process, how to use the resource services depends on these dependencies. This issue will be further investigated in the future.

REFERENCES

[1] J. Wan, B. Chen, M. Imran, F. Tao, D. Li, C. Liu, and S. Ahmad, "Toward dynamic resources management for IoT-based manufacturing," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 52–59, Feb. 2018.

[2] Y. Wu, F. He, D. Zhang, and X. Li, "Service-oriented feature-based data exchange for cloud-based design and manufacturing," *IEEE Trans. Services Comput.*, vol. 11, no. 2, pp. 341–353, Mar./Apr. 2018.

[3] H. Li, M. Liang, and T. He, "Optimizing the composition of a resource service chain with interorganizational collaboration," *IEEE Trans. Ind. Informat.*, vol. 13, no. 3, pp. 1152–1161, Jun. 2017.

[4] H. Li and T. He, "Selecting key feature sequence of resource services in industrial Internet of Things," *IEEE Access*, vol. 6, pp. 72152–72162, Dec. 2018.

[5] H. Li, K. C. C. Chan, M. Liang, and X. Luo, "Composition of resource-service chain for cloud manufacturing," *IEEE Trans. Ind. Informat.*, vol. 12, no. 1, pp. 211–219, Feb. 2016.

[6] B. Cheng, M. Wang, S. Zhao, Z. Zhai, D. Zhu, and J. Chen, "Situation-aware dynamic service coordination in an IoT environment," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2082–2095, Aug. 2017.

[7] Y. Zhang, Z. Guo, J. Lv, and Y. Liu, "A framework for smart production-logistics systems based on CPS and industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 14, no. 9, pp. 4019–4032, Sep. 2018.

[8] N. Moustafa, E. Adi, B. Turnbull, and J. Hu, "A new threat intelligence scheme for safeguarding industry 4.0 systems," *IEEE Access*, vol. 6, pp. 32910–32924, 2018.

[9] Y. Li, Z. Sun, L. Han, and N. Mei, "Fuzzy comprehensive evaluation method for energy management systems based on an Internet of Things," *IEEE Access*, vol. 5, pp. 21312–21322, 2017.

[10] L. Catarinucci, D. de Donno, L. Mainetti, L. Palano, L. Patrono, M. L. Stefanizzi, and L. Tarricone, "An IoT-aware architecture for smart healthcare systems," *IEEE Internet Things J.*, vol. 2, no. 6, pp. 515–526, Dec. 2015.

[11] B. R. Ferrer and J. L. M. Lastra, "Private local automation clouds built by CPS: Potential and challenges for distributed reasoning," *Adv. Eng. Inform.*, vol. 32, pp. 113–125, Apr. 2017.

[12] S. Kim and S. Park, "CPS (cyber physical system) based manufacturing system optimization," *Procedia Comput. Sci.*, vol. 122, pp. 518–524, Nov. 2017.

[13] H. Li and Y. Gao, "Mining QoS benchmark of resource-service chain for collaborative tasks," *Int. J. Internet Manuf.*, vol. 5, nos. 2–3, pp. 121–136, May 2018.

[14] B. Cheng, C. Li, S. Zhao, and J. Chen, "Semantics mining & indexing-based rapid Web services discovery framework," *IEEE Trans. Services Comput.*, to be published.

[15] Q. Wu, F. Ishikawa, Q. Zhu, and D.-H. Shin, "QoS-aware multigranularity service composition: Modeling and optimization," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 11, pp. 1565–1577, Nov. 2016.

[16] M. Chen, W. Saad, and C. Yin, "Virtual reality over wireless networks: Quality-of-service model and learning-based resource management," *IEEE Trans. Commun.*, vol. 66, no. 11, pp. 5621–5635, Nov. 2018.

[17] X. Min, X. Xu, Z. Liu, D. Chu, and Z. Wang, "An approach to resource and QoS-aware services optimal composition in the big service and Internet of Things," *IEEE Access*, vol. 6, pp. 39895–39906, Jul. 2018.

[18] S. Ding, X. He, and J. Wang, "Multiobjective optimization model for service node selection based on a tradeoff between quality of service and resource consumption in mobile crowd sensing," *IEEE Int. Things J.*, vol. 4, no. 1, pp. 258–268, Feb. 2017.

[19] J. Zhao and X. Wang, "Two-sided matching model of cloud service based on QoS in cloud manufacturing environment," *Comput. Integr. Manuf.*, vol. 22, no. 1, pp. 104–112, Jan. 2016.

[20] S. Zhang and H.-B. Li, "Resource selection method based on workflow in cloud manufacturing," *Comput. Intergrated Manuf.*, vol. 21, no. 3, pp. 831–839, Mar. 2015.

[21] V. Gabrel, M. Manouvrier, K. Moreau, and C. Murat, "QoS-aware automatic syntactic service composition problem: Complexity and resolution," *Future Gener. Comput. Syst.*, vol. 80, pp. 311–321, Mar. 2018.

[22] Y. Sun, J. White, B. Li, M. Walker, and H. Turner, "Automated QoS-oriented cloud resource optimization using containers," *J. Automated Softw. Eng.*, vol. 24, no. 1, pp. 101–137, Mar. 2017.

[23] K. Alhamazani, R. Ranjan, P. P. Jayaraman, K. Mitra, C. Liu, F. Rabhi, D. Georgakopoulos, and L. Wang, "Cross-layer multi-cloud real-time application QoS monitoring and benchmarking as-a-service framework," *IEEE Trans. Cloud Comput.*, vol. 7, no. 1, pp. 48–61, Jan./Mar. 2019.

[24] V. R. Chandakanna and V. K. Vatsavayi, "A QoS-aware self-correcting observation based load balancer," *J. Syst. Softw.*, vol. 115, pp. 111–129, May 2016.

[25] Y. Chen, S. Tang, N. Bouguila, C. Wang, J. Du, and H. Li, "A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data," *Pattern Recognit.*, vol. 83, pp. 375–387, Nov. 2018.

[26] Y. Chen, S. Tang, L. Zhou, C. Wang, J. Du, T. Wang, and S. Pei, "Decentralized clustering by finding loose and distributed density cores," *Inf. Sci.*, vols. 433–434, no. 4, pp. 510–526, Apr. 2018.

[27] Y. Chen, S. Tang, S. Pei, C. Wang, J. Du, and N. Xiong, "DHeat: A density heat-based algorithm for clustering with effective radius," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 4, pp. 649–660, Sep. 2017.

[28] Y. Sun, J. White, S. Eade, and D. C. Schmidt, "ROAR: A QoS-oriented modeling framework for automated cloud resource allocation and optimization," *J. Syst. Softw.*, vol. 116, pp. 146–161, Jun. 2016.

[29] L. Cheng, Y. Wang, and Y. Zhou, "EADetection: An efficient and accurate sequential behavior anomaly detection approach over data streams," *Int. J. Distrib. Sensor Netw.*, vol. 14, Oct. 2018, Art. no. 1550147718803303.

[30] S. S. Roy, A. Mallik, R. Gulati, M. S. Obaidat, and P. V. Krishna, "A deep learning based artificial neural network approach for intrusion detection," in *Proc. Int. Conf. Math. Comput.* Singapore: Springer, Apr. 2017, pp. 44–53.

[31] S. Takeuchi, T. Nishino, T. Saito, and I. Watanabe, "Semi-supervised approach to soft sensor modeling for fault detection in industrial systems with multiple operation modes," in *Proc. Int. Conf. Adv. Intell. Syst. Inform.* Cairo, Egypt: Springer, Aug. 2017, pp. 683–693.

[32] J. Liu, Y. Zhou, and Y. Zhang, "Wavelet-based approach for anomaly detection of online services in cloud computing systems," *J. Tsinghua Univ. Sci. Technol.*, vol. 57, no. 5, pp. 550–554, May 2017.

[33] S. Seth and N. Singh, "Dynamic threshold-based dynamic resource allocation using multiple VM migration for cloud computing systems," in *Proc. Int. Conf. Inf., Commun. Comput. Technol.*, Oct. 2017, pp. 106–116.

[34] A. I. Jehangiri, R. Yahyapour, E. Yaqub, and P. Wieder, "Distributed predictive performance anomaly detection for virtualised platforms," *Int. J. High Perform.*, vol. 11, no. 4, pp. 279–290, Jul. 2018.

[35] O. Ibidunmoye, A. R. Rezaie, and E. Elmroth, "Adaptive anomaly detection in performance metric streams," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 217–231, Mar. 2018.

[36] S. Garg, K. Kaur, N. Kumar, and J. J. P. C. Rodrigues, "Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 566–578, Mar. 2019.

[37] L. Li, S. Li, and S. Zhao, "QoS-aware scheduling of services-oriented Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1497–1505, May 2014.

[38] T. Walther, "Digital transformation of the global cement industry," in *Proc. IEEE-IAS/PCA Cement Ind. Conf. (IAS/PCA)*, May 2018, pp. 1–8.

HAIBO LI received the Ph.D. degree from the School of Computer Science and Technology, Harbin Institute of Technology, in 2008.

He is currently a Professor with the College of Computer Science and Technology, Huaqiao University, Xiamen, China. He was a Research Scholar at Hong Kong Polytechnic University, Hong Kong, China, from September 2014 to March 2015. His research interests include in the areas of service-oriented manufacturing systems, such as cloud manufacturing and manufacturing service management, intelligent optimization theory, and algorithm. He is the author or coauthor of more than 40 publications and has received several awards for his research. His published works appear in several refereed journals, including the *Journal of Software*, the *Chinese Journal of Computers*, and *Computer Integrated Manufacturing System*.

JUNCHENG TONG received bachelor's degree from Tianjin Agricultural University, in 2017. He is currently pursuing the M.S. degree with the College of Computer Science and Technology, Huaqiao University, Xiamen, China. His main research interests include in the areas of service science and big data analytics.

SHAOYUAN WENG received the bachelor's degree from the Quanzhou University of Information Engineering, in 2018. She is currently pursuing the M.S. degree with the College of Computer Science and Technology, Huaqiao University, Xiamen, China. Her main research interests include in the areas of analysis of data mining and collaborative task.

XINMIN DONG received the bachelor's degree from Sanming University, in 2019. He is currently pursuing the M.S. degree with the College of Computer Science and Technology, Huaqiao University, Xiamen, China. His main research interests include in the areas of data mining and collaborative task.

TING HE was born in 1972. He received the B.S. degree in mechanical design and manufacturing, and the M.S. degree in management engineering from the Harbin Institute of Technology, in 1995 and 1997, respectively, and the Ph.D. degree from the Department of Mechatronical Engineering, Harbin Institute of Technology, in 2000.

He worked with the School of Computer Science and Technology, Harbin Institute of Technology, from 2000 to 2016. He is currently a Professor with the College of Computer Science and Technology, Huaqiao University. His main research interest includes smart computing and services. He is also exploring the data-enabled innovation of business model and intelligent/service oriented manufacturing.