Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 6629449, 14 pages
https://doi.org/10.1155/2021/6629449

Research Article

Fuzzy Rule-Based Trust Management Model for the Security of Cloud Computing

Mona Soleymani,[1] **Navid Abapour**,[2] **Elham Taghizadeh**,[3] **Safieh Siadat**,[4] **and Rasoul Karkehabadi**[5]

[1]*Department of Computer Engineering, Islamic Azad University, Parand Branch, Tehran, Iran*
[2]*Department of Computer Science, Faculty of Science, University of Mohaghegh Ardabili, Ardabil, Iran*
[3]*Wayne State University, 4815 Fourth Street, Detroit 48202, MI, USA*
[4]*Department of Computer Engineering and Information Technology, Payame Noor University (PNU), P.O. Box 19395-4697, Tehran, Iran*
[5]*Department of Computer Science and Information Technology, Mahdishahr Branch, Islamic Azad University, Mahdishahr, Iran*

Correspondence should be addressed to Safieh Siadat; safieh.siadat@gmail.com

Received 27 December 2020; Revised 25 May 2021; Accepted 30 May 2021; Published 15 June 2021

Academic Editor: Ali Asghar Rahmani Hosseinabadi

In the last few years, due to the benefit of solving large-scale computational problems, researchers have been developed multicloud infrastructures. The trust-related issue in multiclouds includes more complicated content and new problems. A new trust management framework for multicloud environments is proposed in this article. The proposed framework used a combination of objective and subjective trust values to calculate the cloud service provider's trust values. This new framework can identify and rectify fake feedbacks from other feedbacks. Another advantage of this framework is applying fuzzy rules to calculate trust values. Two main components of the proposed framework are simulated in this paper. The simulation results confirm the important role of applied components. Also, this paper proposed a framework compared with other frameworks (feedback-based model, SLA-based model, and multicloud model). Simulation results show the proposed framework increased trust values rather than other models. Also, compared with other models, our framework gives better mean trust values.

1. Introduction

In recent years, cloud computing has attracted the attention of many researchers around the world, and various programs, infrastructures, and frameworks have been created for it by several companies in the world [1–4]. In fact, cloud computing, as a new technology, provides a fully scalable, accessible, and flexible computing platform for a variety of applications [5]. Due to the various applications that cloud computing has found in various aspects of life, the issue of providing security in cloud computing communications and data stored in it, has been considered by users and providers of cloud computing services. According to some research conducted at Berkeley University, trust management and security optimization have been identified as the most important issues in using various cloud computing services [3, 6–8]. Cloud computing due to its distributive nature, very dynamic space, and lack of transparency in performing cloud computing faces many challenges in providing security and gaining trust. In order to improve security in performing cloud computing, trust management can play a very effective role [9, 10].

This article is organized into five sections. In the second part, we examine some of the related work done by various researchers. In the third section, we present a new framework for trust management in multicloud environments. In the fourth section, we bring the simulation results of the framework presented in this research, and finally, in the fifth section, we will conclude.

In this paper, we have tried to present a trust management model in multicloud environments that uses both objective and subjective parameters to calculate trust values. And it looks at the category of trust instead of a one-dimensional category in a multidimensional way and selects and assigns the cloud service provider that is closest to the cloud service user request. In this model, an attempt has been made to design a component to detect fake feedback and distinguish fake feedback from the real one and use only real feedback to calculate trust values, which leads to increasing the accuracy of the proposed framework, so, in this paper, for the first time, the calculation of multidimensional trust values in multicloud environments is presented along with the detection of fake feedback.

In previous models and methods, trust has been considered as a one-dimensional category, whereas trust is a multidimensional and relative value, and for each individual, some parameters of service quality are important. Someone needs to have accessibility, someone else reliability, the other security, and so on. And after receiving the service, the feedback will also vary according to the importance of each of these parameters. Therefore, trust cannot be viewed as a one-dimensional parameter, and it should be calculated multidimensional, and in this model, based on the type of user's request and prioritizing the parameters of the cloud service provider that is closest to the request, it is selected. And also, after receiving the service despite the appropriate cloud service, users may generate fake feedbacks in order to reduce the trust values of cloud service providers, in the proposed framework which has been tried to provide a component for detecting fake feedbacks in this component with higher accuracy and calculate the trust values. So, in general, it can be stated that in other previous studies, the issue of trust was viewed as a one-dimensional way, and besides there was no appropriate mechanism to detect fake feedback in this study for the first time.

2. Related Works

The level of trust and confidence in cloud service providers is one of the important parameters to provide a reliable service for the cloud service user. Liu and colleagues [11] proposed a method in which reliable cloud service providers for SaaS applications were selected based on their credibility and trust [12]. Many of the proposed models for measuring trust are based on records of trust in various cloud service providers [13, 14]. Accordingly, these models can be divided into two general categories, which are subjective and objective trust models [15].

To measure the level of objective trust, parameters related to the quality of service (QoS) delivery are used. Fan and colleagues [16] proposed a concept called "objective trust" for software agents. The researchers explained the trust between agents based on real-life experiences. Lin et al. [17] proposed a new framework for MANET networks in which one node evaluates the reliability of another node using direct observations. To calculate the level of trust in the subjective method, we can use the amount of feedback received from users using various cloud services [16]. Uikey et al. [18] proposed a new trust management model in which all information about different cloud service providers and the level of trust is recorded and stored. In this study, SLA models were used to calculate the reliability of cloud service providers. Alhamad et al. [3] proposed a new SLA-based trust management model to predict the level of trust in cloud service providers. In the proposed model, SLA-based conceptual framework is integrated with a trust value management. Chakraborty et al. [19] applied the parameters extracted using the SLA to measure the reliability of cloud services. Some of the most commonly used SLA-based models are probability-based trust model, intuitive reasoning-based trust model [20], Bayesian-based trust model, Dempster–Shafer model, Fuzzy logic-based trust [21], cloud computing trust model [22], and so on. Siadat et al. [23] proposed a new model for managing trust in cloud computing that uses game theory to detect fake feedback [24]. Chen et al. [25] proposed a new trust management model for the Internet of Things (IoT) in which trust management at different levels of the IoT was examined. In the study by Guo et al. [26], a new model for managing trust in the IoT suggests that methods for assessing trust are examined based on five common design dimensions (including trust composition, dissemination, aggregation, updating, and shaping). Din et al. [27] examined trust management methods without performing any classification. Various studies have been conducted to combine methods of objective and subjective trust. Yuan et al. [28] proposed a framework for assessing trust that uses a combination of objective and subjective trust methods that calculate and rate trust based on a combination of users' trust and credibility. Ngo et al. [29] examined the relationship between the level of objective trust and subjective trust and expressed the characteristics of each of these two types of trust.

Sangaiah et al. [30] with using machine-learning techniques proposed a new method to maintain the confidentiality of the geographical location of PBS portable users. The proposed method had three phases. During these phases, using the integration of decision tree techniques and the nearest neighbor, the user's geographical location was determined, and using the sequence of routes transferred and using hidden Markov models, the user's destination was identified. Along with maintaining the confidentiality of users' position, these researchers showed that the accuracy of this method in establishing position in PBS was equal to 90%. The results of the implementation of the proposed method by these researchers showed that the accuracy of this method in establishing position confidentiality in PBS was equal to 90%.

Sangaiah et al. [31] defined a weight called relay ability for each node according to the sensor network topology. These weights are calculated by the head and reported to all sensor nodes. When a target enters the area covered by sensor nodes, a signal is sent to CH via a path that has a predefined maximum weight in the network. The simulation of the proposed method in this research showed that this method has better results than other tracking methods based on the criteria of network power consumption, power consumption and power for GRTT, dynamic energy efficient

routing (DEER) protocol, and virtual power-based energy consumption (VFEM).

In the study by Sangaiah et al. [32], an energy-aware green adversary model has been proposed for use in intelligent industrial environments by achieving confidentiality. In this study, researchers explored various aspects of preserving geographic location information and information confidentiality. Finally, we proposed a new model which has the capacity to make prediction based on a schedule in real-time situations, it can make connections, respond to user demands, and minimize energy consumption. The experimental results of these researchers showed that their proposed model can be five times more energy saving compared with other methods.

Mousa et al. [33] used a trust model based on the fuzzy logic system to evaluate trust values. They proposed a new method for this purpose, which gives cloud users the ability to assess the reliability of cloud service providers. Simulations of their proposed method showed that the accuracy of the evaluations performed by these researchers was higher compared with other works.

Sule et al. [34] using a combination of fuzzy logic and several different security mechanisms (such as identification and trust) proposed a multilayered security model based on an integrated cloud platform. The simulation of the proposed model showed that this model can provide the possibility of determining and verifying the trust status of the cloud computing service. Therefore, this model can be used to improve end-user confidence when selecting or consuming cloud computing resources.

Fan et al. [35] assessed the objective and subjective trust of CSPs. Using different clouds, they proposed a trust dissemination network among TSPs. This network can be used by the TSP to obtain trust information about a service from other TSPs. The researchers also proposed a framework for trust management in a multicloud environment based on a trust assessment and a proposed trust dissemination network. The results of their experiments showed that the proposed framework is more reliable than other CSPs in a multicloud environment.

In the study by Kumar et al. [36], a fuzzy-based trust management system was proposed to facilitate cloud service delivery and identify trusted providers. The proposed system simulation showed that the degree of reliability and trust of the proposed system was higher compared with that of other CSPs.

Over the past few years, numerous studies have been conducted on multicloud infrastructure and cloud computing environments to solve large-scale computing problems. The problems associated with building and managing trust across multiple clouds are vast. Nielsen et al. [37] proposed a new framework for trust management in multicloud environments for cloud service providers. Among the models offered for trust management, only a few of them have considered trust in a multidimensional way, and for this purpose, they have combined objective and subjective trust. Also, a limited number of these models are able to detect fake feedback models. The main purpose of this paper is to provide a framework for trust management in multicloud environments using fuzzy logic to enhance security in these networks. The research questions that this research follows are how can trust management help to enhance security in multicloud networks? Is it possible to calculate trust values from the combination of subjective parameters and objective parameters together using fuzzy logic? How can we distinguish fake feedback from nonfake feedbacks in the proposed framework to increase the accuracy of trust values?

3. Our Proposed New Trust Management Framework in Multicloud Environments

We propose a new framework for trust management using cloud service providers (TSPs). We try to cover several problems that exist in the field of trust management in multicloud environments with this proposed framework. The trust management framework proposed in this study is seen in Figure 1.

CSPs (cloud service providers) are responsible for providing services to cloud service users. In cloud computing, a variety of services are provided to users by CSPs. The most common types of services are SaaS, PaaS, and IaaS. CSPs also provide services to CSUs (cloud service users). CSUs send their requests to TSP (which is one of the CSPs). The selection of this TSP among CSPs is done by different selection algorithms. The main task of TSP is to select the appropriate CSP to receive and respond to requests sent from CSU. Another function of TSP is to verify CSP reliability.

3.1. SLA Monitor Agent. SLA monitor agent is located on CSUs side. It monitors services behavior and services performance that if CSUs meet SLA or not. SLA monitor agent collects data in the interaction between CSPs and CSUs and also its responses to control requests. A control request is sent by TSP. SLA monitor agent continuously collects control information from server side. Control information contains SLA performance parameters.

3.2. Monitoring Information Collection Agent. The responsibility of monitoring information collection agent is collecting monitor agents information on SLA which has an agreement with TSP. Collected information by this agent is applied to evaluate objective trust values. The following information is maintained by the monitoring information collection agent:

(i) CSPs list is monitored by TSP

(ii) SLA monitoring information received from SLA monitor agent that is in agreement with TSP

Before receiving the CSUs service from the CSPs, an SLA contract is agreed between them with various parameters. This SLA contract is the output of the SLA negotiation component, which determines the level of service that the CSUs and CSPs agree on. Some of the parameters in the SLA are availability, response time, and so on, which are agreed upon, based on which the server agreements that are closest to the CSUs request are selected in the next steps.

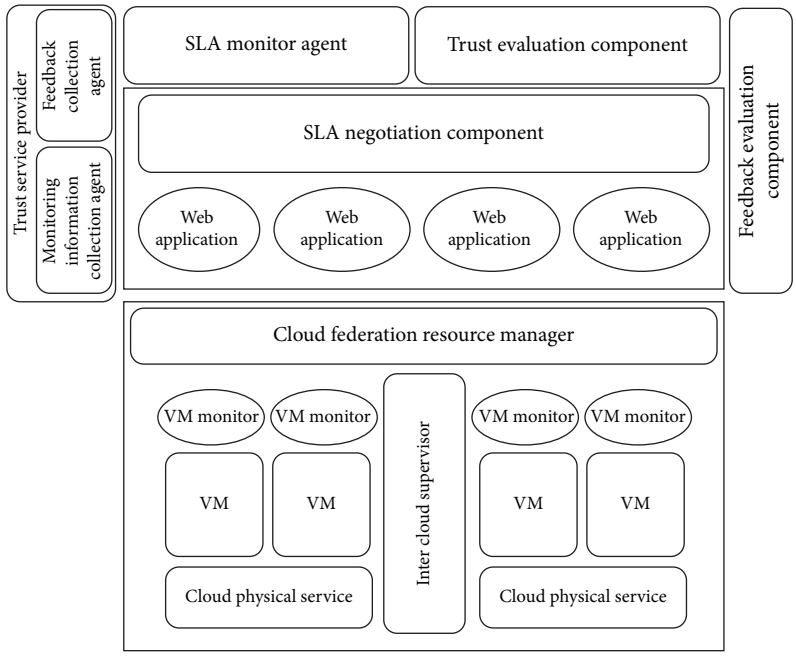

Figure 1: Our proposed trust management framework in multicloud environments.

3.3. SLA Negotiation Component. SLA negotiation component negotiates between CSUs and CSPs. SLA with multiple CSPs negotiation is a complicated work. Usually, the third part does this task. In this paper, the name of this third part is the SLA negotiation component. Figure 2 illustrates the SLA negotiation component.

SLA negotiation component includes the following agents:

(i) Registry agent
(ii) Provider agent
(iii) Negotiation agent
(iv) Trust assessment agent to requests submitted by users
(v) Mediator agent
(vi) Client agent

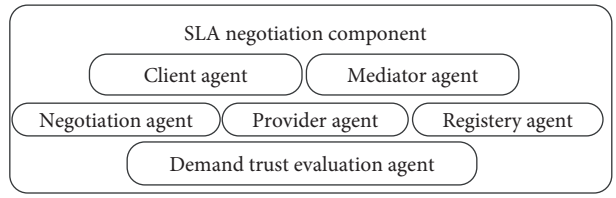

Figure 2: SLA negotiation component.

3.3.1. Registry Agent. The task of REA is to register CSPs information together with their services.

3.3.2. Provider Agent. PA represents CSPs. All negotiations between CSUs and CSPs do by their representatives. Negotiations such as SLA agreement and cost agreement are done by the provider agent.

3.3.3. Negotiation Agent. NA task is SLA production and SLA optimization. CSPs register their services by NA. CSUs provide CSPs SLA details by NA. After successful negotiation between CSP and CSU, they register the negotiation process to investigate and future validation. Negotiation process registers if negotiation between CSP and CSU is finished or not. NA registers negotiation process at specific period of time.

Stored information by NA is as follows:

(i) One directory of CSPs services (CSP may provide multiple services on multiple clouds but for simplicity assumed that CSP provides one service in multiple clouds)
(ii) A catalog of service request from CSP
(iii) In each CSU, service negotiation includes negotiation requester, negotiation start and end time, negotiation content, and negotiation result

3.3.4. Demand Trust Evaluation Agent. This agent is responsible for receiving requests sent from users and sends these requests to the best available server based on the type of service requested. To achieve this goal, the parameters related to the evaluation of the quality of services (QoS) provided to users in the platform are initialized. This agent selects the server that best matches the services requested by the user to respond to the reliability assessment factor to the requests submitted by the user in batches. This factor performs a two-step evaluation of the reliability of the requests submitted by users:

(i) Trust evaluation using the parameters provided in the user-submitted request

(ii) Select the best server to respond to requests and provide the requested services optimally and assign the request to that server

The various parts of the reliability assessment agent to the requests submitted by users are seen in Figure 3.

(1) Trust Evaluation with Demand Parameter Level. In this section, to evaluate the level of trust in the user-submitted requests, several servers are selected as candidates to provide the user requested services. Several parameters affect the quality of services (QoS) provided, which are as follows:

(i) Duration of delay in providing requested services

(ii) Duration of response to the request

(iii) Accuracy of providing service

(iv) The amount of cost requested to provide the requested services of users

Each user sets an initial value for each of the mentioned parameters based on their preferences. To evaluate the amount of trust using the mentioned parameters, it is calculated by accessing the trust repository. For each calculation, the cloud service providers denoted by p have the maximum value defined for reliability and are selected as candidates. Requests submitted by users will be sent to these candidates. Formulas (1) to (5) include all of the steps mentioned above.

$$\mathrm{DP} = (\mathrm{dp}_1, \ldots, \mathrm{dp}_m), \tag{1}$$

$$w_i = \frac{\mathrm{dp}_i}{\sum_{i=1}^{m} \mathrm{dp}_i}, \quad \sum w_{i=1}, \tag{2}$$

$$\mathrm{dtv}_i = \sum_{i=1}^{n} \sum_{j=1}^{m} w_j, p_{j,i}, \tag{3}$$

$$\mathrm{DTV}_j = (\mathrm{dtv}_1, \mathrm{dtv}_2, \mathrm{dtv}_3, \ldots, \mathrm{dtv}_n). \tag{4}$$

$$\mathrm{DTV} = \begin{bmatrix} \mathrm{DTV}_1 \\ \mathrm{DTV}_2 \\ \mathrm{DTV}_3 \\ . \\ . \\ \mathrm{DTV}_k \end{bmatrix}. \tag{5}$$

In formula (1), the DP shows a list of parameters that have been initialized by the user. In formula (2), w_i represents the weight assigned to each parameter by the user. In formula (3), dtv_i shows the amount of trust in the requests provided by the service providers and m indicates the number of parameters used. Formula (4) uses a dtv_i storage for DTV_j for each request and n the number of source nodes. In formula (5), DTV is an array of DTV_j while k is the size of each batch of requests. We calculate and save DTVs in batches.

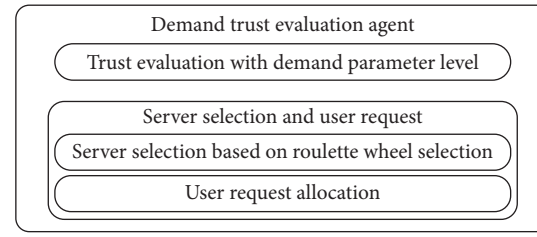

Figure 3: Trust assessment agent to requests submitted by users.

(2) *Server Selection and User Request.* In this section, the most appropriate cloud service provider is selected and requests are sent from users to the selected service provider. In this part, two series of operations are performed as follows:

(i) *Selecting the Most Suitable Server Based on the Roulette Wheel Mechanism.* In this research, the roulette wheel mechanism is used to select the best and most appropriate service provider. The reason for using this mechanism is to create a load balance among all cloud service providers. Formulas (6) to (10) calculate the number of user requests (in percentage) allocated to each cloud service provider. In formula (6), m represents the number of parameters used, w represents the weight assigned to each parameter, and P_i represents the value of the parameter stored in the trust repository. In formula (7), w represents the weight assigned to each parameter. In formula (8), the value of dtv_i is stored in the T.V array. Sp_i is the percentage of user requests assigned to the i server in formula (9). Finally, in formula (10), the value of Sp_i is stored in the SP array.

$$w = \frac{1}{m}, \tag{6}$$

$$\mathrm{t.v}_i = \sum_{i=1}^{n} \sum_{j=1}^{m} w \times p_{j,i}, \tag{7}$$

$$\mathrm{T.V} = (\mathrm{tv}_1, \mathrm{tv}_2, \ldots, \mathrm{tv}_n), \tag{8}$$

$$\mathrm{sp}_i = \frac{\mathrm{tv}_i}{\sum_{i=1}^{n} \mathrm{tv}_i}, \tag{9}$$

$$\mathrm{SP} = (\mathrm{sp}_1, \mathrm{sp}_2, \ldots, \mathrm{sp}_n), \quad \sum_{i=1}^{n} \mathrm{sp}_i = 1. \tag{10}$$

(ii) *Assigning the User Request to the Selected Server.* In this section, using SP and roulette wheel mechanism, we will select the best server from the candidate servers selected in the previous step so that we can assign the request submitted by the user to it.

3.3.5. Mediator Agent. Mediator agent (MA) extracts available CSP from RA. MA sends available CSPs list to demand trust evaluation agent (DTEA). MA contacts

between client agent and provider agent. MA determines services price by this contact. MA transfers selected CSPs by CSP selecting agent and load balancing agent with prices to the client agent. The client agent selects one of CSP among CSPs. MA sends selected CSP to the negotiation agent.

3.3.6. Client Agent. The client agent receives the service request of CSU. The service request of CSU has contained QOS parameters. Client agent sends service request of CSU to MA.

3.4. Feedback Evaluation Component. A new feedback evaluation component has been presented in this paper. This component evaluates and updates the received feedback from CSU after receiving service. It qualitatively identifies and rectifies fake feedback. This new component also prevents circumvention, collusion, latency, and impersonation attacks.

(i) *Feedback*. Feedback means the text or reaction that the user sends to the server after receiving the requested service. This feedback can be used to predict the reliability of CSPs. Some attacks are carried out by hackers to change the content of the feedback, which makes the CSPs less reliable. Some of these threats against feedback include circumvention, collusion, delaying, and impersonation.

(ii) *Circumvention*. When CSU requests a regular service from the cloud service provider and after receiving the service, it sends a negative feedback to the server. In this case, the CSU reduces the amount of trust in the CSP with this feedback. To avoid reducing trust in CSP, we identify this type of feedback as fake feedback.

(iii) *Collusion*. Collusion has occurred when several unauthorized and malicious users unite and attack a single cloud service provider. The method of attack of these users is malicious, by sending fake feedback to service providers. This type of attack is very difficult to detect using traditional methods.

(iv) *Delay*. When some CSUs repeatedly send a request to receive a service that is very time consuming to respond, it is said that the attack has been delayed. CSU sends negative feedback based on the delay. CSUs are forced to waste a lot of time servicing those requested services. Therefore, they will not be able to serve other CSUs, and eventually there will be a delay. Identifying this type of malicious user using classical methods is very difficult or sometimes impossible.

(v) *Impersonation*. In several cases, some users may impersonate other users and send fake feedback instead. To combat these threats, CSU checks users' identities and performs authentication operations for them. In addition, the role of each user in the interactions performed and the requested or received services must be confirmed.

The components used to evaluate the feedback are shown in Figure 4, which are as follows:

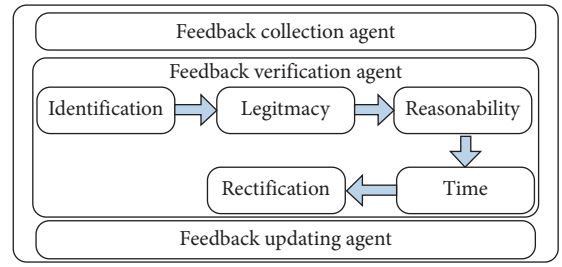

Figure 4: Feedbacks evaluation component.

(i) An agent for collecting feedback

(ii) An agent to confirm the feedback received

(iii) The agent of updating the feedback received

3.4.1. Feedback Collection Agent. The agent collects feedbacks and sends them to the verification of received feedback agent.

3.4.2. Feedback Verification Agent. This agent is responsible for identifying and correcting negative feedback. Operations in this agent include identifying negative feedback, legitimizing that feedback, reasoning, timing, and correcting that feedback.

(1) Identification. To identify negative feedback, we must first identify the credibility of the user who requested the cloud service. All CSUs have a unique identifier and must register their actual details when submitting a request. The feedback agent must verify this ID. In this case, we first check the CSU ID that sends the feedback. If the CSU ID is verified, the feedback will be labeled "valid"; otherwise, it will be identified as "fake feedback." Fake feedback is immediately discarded or removed.

(2) Legitimacy. The term legitimacy refers to the validity or value of the feedback received. To check the legitimacy, we need to check the unique CSU ID and the CSP ID during the transaction. Legitimacy prevents unauthorized users from infiltrating the cloud and prevents these unauthorized users from being able to send feedback over the network.

(3) Time Threshold. An evaluation and verification factor considers a threshold period after the transaction to receive feedback from the client. If feedback is received before or after the defined time limit, it will be ignored or deleted immediately.

(4) Reasonability. One of the most complex aspects of feedback validation is that it is logical as there is no set point for evaluating it. To identify and validate the logic of the feedback received, we compare the current feedback with previous feedback or feedback. If the difference between these feedbacks is less than Delta, the feedbacks are considered reasonable; otherwise, they are either ignored or need to be corrected.

(5) Rectify. If the previous methods (detection of negative feedback, use of legitimacy, or time limit) confirm that the feedback is fake, that feedback will be ignored and discarded. However, if the review of feedback shows that it is logically problematic, we should try to correct that feedback. Formulas (11) to (14) are used to correct fake feedback.

$$a_i = \frac{1}{m}\sum_{k=1}^{m} f_{\mathrm{CSP}_{i,k}}, \tag{11}$$

$$m = \min\left(m, \mathrm{feedbackcountCSP}_i\right), \tag{12}$$

$$F'_{\mathrm{CSP}_i} = a = \left(f_{\mathrm{CSP}_i} - a\right) \times r_{\mathrm{CSU}_0}, \tag{13}$$

$$r_{\mathrm{CSU}_0} = \frac{r_{\mathrm{CSU}_j}}{\sum_{k=1}^{n} r_{\mathrm{CSU}_k}}. \tag{14}$$

In formula (11), a_i presents the previous m feedback average, $f_{\mathrm{CSP}_{i,k}}$ is the CSP_i feedback, and the k index represents the previous k feedback of CSP_i. m is obtained from formula (12). Feedback count CSP_i is the number of received feedback from CSP_i

In formula (13), F'_{CSP_i} is the rectified feedback of CSP_i and r_{CSU_0} can be obtained from formula (14). In formula (14), r_{CSU_j} is the *jth* CSU reliability and n represents the number of cloud service users. When CSU_j sends true feedback, r_{CSU_j} increases, and when it sends fake feedback, r_{CSU_j} decreases. The algorithm of the feedback verification agent is shown in Figure 5.

3.4.3. Feedback Updating Agent. The role of this agent is to update the feedback received from the feedback confirmation agent.

3.5. Trust Evaluation Component. The responsibility of the trust assessment component is to calculate the amount of subjective trust and the amount of objective trust. To calculate the amount of subjective trust and the amount of objective trust from the information collected, we use the supervised data collection agent and the feedback evaluation component. After calculating the amount of trust, another task of the trust assessment component is to update the amount of trust stored in the trust repository. The component of trust assessment includes the following factors:

(i) An agent to evaluate the amount of subjective trust

(ii) An agent for evaluating the amount of objective trust

(iii) An agent for updating the amount of trust

Figure 6 shows the trust evaluation component.

Trust repository includes CSPs trust values that they calculated and stored. Trust values will be used as a trust evidence for future decisions.

Trust modeling methodology assumptions are as follows:

(i) Each CSP only can present one type of service

(ii) Each CSU uses the terms Trust (T), Distrust ($-T$), and Uncertain (T, $-T$) for trust values that they obtain by fuzzy rules after calculating trust values

```
1 feedback verification agent (receives feedback from
  feedback collection agent)
2 Begin
3 for i = 1 to m do // m = min (m, feedback_count_CSP)
                        a = (1/m) Σ_{i=1}^{l} f_{CSP_i}
4
      (f_{CSP_i} − a > δ)
5 If                    then //else don't do anything
6     rectify ()
7 End
```

Figure 5: Feedbacks verification agent.

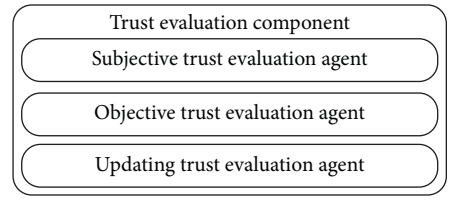

Figure 6: Trust evaluation component.

3.5.1. Subjective Trust Evaluation Agent. Subjective trust values are calculated based on revived feedbacks. The responsibility of subjective trust values agent is receiving CSPs feedbacks that exist in their domain. Subjective trust values evaluation agent receives the feedbacks from the feedback evaluation component, and also it receives the parameters of requested service with their weights from the trust negotiation component. Subjective trust values evaluation agent calculates CSU service satisfaction by using received feedbacks and requested service parameters. CSU service satisfaction is calculated by using formulas (15) to (20).

$$w = (w_1, w_2, \ldots, w_m), \tag{15}$$

$$F' = \sum_{i=1}^{m} F_{p_i} \times w_i. \tag{16}$$

In formula (15), we have trust negotiation component. In formula (16), F_{pi} is received from feedbacks evaluation component. F_{p_i} are feedbacks of each parameter. W_i in formula (16) is obtained from formula (2) and m is the number of parameters.

$$\forall i, \quad i = 1, \ldots, m, P_{dei} = F_{pi}. \tag{17}$$

Formula (18), calculate CSU service satisfaction w_i and p_{dmi} get from trust negotiation component and p_{dei} obtains from formula (17).

$$\overline{f}_{i,j} = \sum_{i=1}^{k} w_i \frac{\left|p_{dmi} - p_{dei}\right|}{p_{dmi}}. \tag{18}$$

In formula (19), $\mathrm{st}^t_{i,j}\{T\}$ is subjective trust value of ith service requested related to CSP_j in time t. $\mathrm{st}^t_{i,j}\{T\}$ is subjective trust value in trust state, $\mathrm{st}^t_{i,j}\{-T\}$ is subjective trust value in distrust state, and $\mathrm{st}^t_{i,j}\{\mathrm{U}\}$ is subjective trust value in uncertain state.

$$\mathrm{ST}_{i,j}^{t} = \begin{cases} \mathrm{st}_{i,j}^{t}\{T\} = \dfrac{\bar{f}_{i,j}^{t} - 0.5}{0.5}, & \text{if } \bar{f}_{i,j}^{t} > 0.5, \\ \mathrm{st}_{i,j}^{t}\{-T\} = \dfrac{0.5 - \bar{f}_{i,j}^{t}}{0.5}, & \text{if } \bar{f}_{i,j}^{t} < 0.5, \\ \mathrm{st}_{i,j}^{t}\{U\} = 1 - \mathrm{st}_{i,j}^{t}\{T\} - \mathrm{st}_{i,j}^{t}\{-T\} & \text{else}, \end{cases} \tag{19}$$

$$\mathrm{LST}_{i,j}^{t} = \begin{cases} \left\{ \mathrm{lst}_{i,j}^{t}\{T\} = \mu \times \mathrm{st}_{i,j}^{t-1}(T) + (1-\mu) \times \dfrac{\bar{f}_{i,j}^{t} - 0.5}{0.5}, \right. \\ \mathrm{lst}_{i,j}^{t}\{-T\} = \mu \times \mathrm{st}_{i,j}^{t-1}(-T) + (1-\mu) \times \dfrac{\bar{f}_{i,j}^{t} - 0.5}{0.5}, \\ \mathrm{lst}_{i,j}^{t}\{U\} = 1 - \mathrm{lst}_{i,j}^{t}\{T\} - \mathrm{lst}_{i,j}^{t}\{-T\}, \\ 0 <= \mu <= 1. \end{cases} \tag{20}$$

In formula (20), LST is the local subjective trust value. $\mathrm{lst}_{i,j}^{t}\{T\}$ is local subjective trust value in trust state, $\mathrm{lst}_{i,j}^{t}\{-T\}$ is local subjective trust value in distrust, state and $\mathrm{lst}_{i,j}^{t}\{\mathrm{U}\}$ is subjective trust value in an uncertain state. In formula (20), μ is the weight factor. If $t = 0$, then the subjective trust value of CSU_i related to CSP_j is zero as follows:

$$\mathrm{LST}_{i,j}^{0} = (0, 0, 1). \tag{21}$$

(1) Calculating Subjective Trust Values. Using Formula (22), global subjective trust (GST) value is calculated; GST_{j}^{t} is CSPj subjective trust value of in time t. $\mathrm{gst}_{j}^{t}\{T\}$ is the global subjective trust value in trust state, $\mathrm{gst}_{j}^{t}\{-T\}$ is the global subjective trust value in distrust state, and $\mathrm{gst}_{i,j}^{t}\{U\}$ is the global subjective trust value in an uncertain state.

$$\mathrm{GST}_{j}^{t} = \begin{cases} \mathrm{gst}_{j}^{t}\{T\} = \dfrac{\sum_{i=1}^{n} \mathrm{lst}_{i,j}^{t}(T)}{n}, \\ \mathrm{gst}_{j}^{t}\{-T\} = \dfrac{\sum_{i=1}^{n} \mathrm{lst}_{i,j}^{t}(-T)}{n}, \\ \mathrm{gst}_{j}^{t}\{U\} = 1 - \mathrm{gst}_{i,j}^{t}\{T\} - \mathrm{gst}_{i,j}^{t}\{-T\}. \end{cases} \tag{22}$$

3.5.2. Objective Trust Values Calculating Agent. CSP ensures a certain level of service performance for CSU that the deal was agreed earlier in the SLA. Services performance will be measured by parameters set. This parameter was proposed by Habib et al. [38].

Based on the study by Habib et al. and different requirements in the industry, many parameters should be investigated (objective trust values parameters). These parameters are availability, reliability, response time, security, privacy, transparency, and consumer protection [39].

Objective trust values evaluate based on measured parameters that whether they met SLA or not. That process is done by an SLA monitoring agent.

Assumptions are as follows:

M is the parameter set used in SLA between CSP and CSU.

$n_{i,j}$ is the number of transactions in w window that t is between 1 and n ($1 \le T <= N$).

For each transaction between CSP and CSU, TSP receives collected records for all parameters of SLA. TSP decides the statuses of those records. The statuses of received service are T, $-T$, and U.

$n_{i,j,k}^{\mathrm{succ}}$ is the transaction final number between CSUi and CSPj based on each k parameters that have been satisfied in SLA.

$n_{i,j,k}^{\mathrm{faillled}}$ is the transaction final number between CSUi and CSPj based on each k parameters that have been dissatisfied in SLA.

$n_{i,j,k}^{\mathrm{UN}}$ is the transaction final number between CSUi and CSPj based on each k parameters that have been located uncertain state in SLA.

In time $t = 0$, there does not exist transaction between CSP and CSU. So, $n_{i,j,k}^{\mathrm{succ}} = 0$, $n_{i,j,k}^{\mathrm{faillled}} = 0$, and $n_{i,j,k}^{\mathrm{UN}} = 0$.

Local objective trust value is calculated by using formulas (23) and (24). Local objective trust value is calculated for window T, and it is calculated based on k parameters from SLA.

In formulas (24) and (25), $\mathrm{LOT}_{i,j,k}^{t}$ is objective trust value of CSUi related to CSPj in time t based on k parameter. μ is weight factor in formulas (24) and (25).

$$\mathrm{LOT}_{i,j,k}^{t} = \left(\mathrm{lot}_{i,j,k}^{t}\{T\}, \mathrm{lot}_{i,j,k}^{t}\{-T\}, \mathrm{lot}_{i,j,k}^{t}\{U\}\right), \tag{23}$$

$$0 < = \mu < = 1,$$

$$\mathrm{LOT}_{i,j,k}^{t} = \begin{cases} \mathrm{lot}_{i,j,k}^{t}\{T\} = \mu * \mathrm{lot}_{i,j,k}^{t-1}(T) + (1-\mu)^{*} \dfrac{n_{i,j,k}^{\mathrm{succ}}}{n_{i,j,k}^{\mathrm{succ}} + n_{i,j,k}^{\mathrm{failled}} + n_{i,j,k}^{\mathrm{UN}}}, \\ \mathrm{lot}_{i,j,k}^{t}\{-T\} = \mu * \mathrm{lot}_{i,j,k}^{t-1}(-T) + (1-\mu)^{*} \dfrac{n_{i,j,k}^{\mathrm{failled}}}{n_{i,j,k}^{\mathrm{succ}} + n_{i,j,k}^{\mathrm{failled}} + n_{i,j,k}^{\mathrm{UN}}}, \\ \mathrm{lot}_{i,j,k}^{t}\{U\} = \mu * \mathrm{lot}_{i,j,k}^{t-1}(U) + (1-\mu)^{*} \dfrac{n_{i,j,k}^{\mathrm{UN}}}{n_{i,j,k}^{\mathrm{succ}} + n_{i,j,k}^{\mathrm{failled}} + n_{i,j,k}^{\mathrm{UN}}}. \end{cases} \tag{24}$$

In time $t = 0$, $\mathrm{LOT}_{i,j,k}^{0} = (0, 0, 1)$.

In formula (25), $\mathrm{LOT}_{i,j,k}^{t}$ is CSUi objective trust value related to CSPj in time t.

$$\mathrm{LOT}_{i,j}^{t} = \begin{cases} \mathrm{lot}_{i,j}^{t}\{T\} = \sum_{k-1}^{m} \omega_k \mathrm{lot}_{i,j,k}^{t}(T), \\ \mathrm{lot}_{i,j}^{t}\{-T\} = \sum_{k-1}^{m} \omega_k \mathrm{lot}_{i,j,k}^{t}(-T), \\ \mathrm{lot}_{i,j,k}^{t}\{U\} = 1 - \mathrm{lot}_{i,j}^{t}\{T\} - \mathrm{lot}_{i,j}^{t}\{-T\}, \end{cases} \tag{25}$$

$$\sum_{k=1}^{m} \omega_k = 1, \tag{26}$$

where $\mathrm{lot}_{i,j}^{t}\{T\}$ is objective trust value probability in trust status from CSUi related to CSPj in time t. $\mathrm{lot}_{i,j}^{t}\{-T\}$ is objective trust value probability in distrust status from CSUi related to CSPj in time t. $\mathrm{lot}_{i,j}^{t}\{U\}$ is objective trust value probability in uncertain status from CSUi related to CSPj in time t.

(2) Calculating Global Objective Trust Values. Global objective trust values for CSPj are obtained by local objective trust values combination. Global objective trust values are obtained from the local objective trust values average that is shown in formula (27). got_{j}^{t} is global objective trust values in time t [21].

$$\mathrm{GOT}_{j}^{i} = \begin{cases} \mathrm{got}_{j}^{t}\{T\} = \dfrac{\sum_{i=1}^{n} \mathrm{lot}_{i,j}^{t}(T)}{n}, \\ \mathrm{got}_{j}^{t}\{-T\} = \dfrac{\sum_{i=1}^{n} \mathrm{lot}_{i,j}^{t}(-T)}{n}, \\ \mathrm{got}_{j}^{t}\{U\} = 1 - \mathrm{got}_{i,j}^{t}\{T\} - \mathrm{got}_{i,j}^{t}\{-T\}. \end{cases} \tag{27}$$

3.5.3. Calculating Trust Values. Trust values will be calculated in the proposed framework by using fuzzy rules that fuzzy inputs are objective trust values and subjective trust values and fuzzy output is trust values.

(i) Fuzzy inputs contain three states: low, medium, and high, and their ranges are between 0 and 1

(ii) Fuzzy outputs contain three states: low, medium, and high, and their ranges are between 0 and 1

Some fuzzy rules are as follows:

$$\begin{aligned} &\mathrm{If}\left(\mathrm{GOT}_{j}^{t}(T) = \mathrm{low}\right) \mathrm{and}\, \mathrm{GST}_{j}^{t}(T) = \mathrm{lo}\, \mathrm{then}\, \mathrm{TV}(T) = \mathrm{low}, \\ &\mathrm{If}\left(\mathrm{GOT}_{j}^{t}(T) = \mathrm{low}\right) \mathrm{and}\left(\mathrm{GST}_{j}^{t}(T) = \mathrm{medium}\right) \mathrm{then}\, \mathrm{TV}(T) = \mathrm{low}, \\ &\mathrm{If}\left(\mathrm{GOT}_{j}^{t}(T) = \mathrm{medium}\right) \mathrm{and}\left(\mathrm{GST}_{j}^{t}(T) = \mathrm{medium}\right) \mathrm{then}\, \mathrm{TV}(T) = \mathrm{medium}, \\ &\qquad \ldots \\ &\qquad \ldots \\ &\mathrm{If}\left(\mathrm{GOT}_{j}^{t}(T) = \mathrm{high}\right) \mathrm{and}\left(\mathrm{GST}_{j}^{t}(T) = \mathrm{high}\right) \mathrm{then}\, \mathrm{TV}(T) = \mathrm{high}. \end{aligned} \tag{28}$$

Trust Values Updating Agent. Trust values updating agent task is updating trust repository. Trust value is obtained by fuzzy rules map in three states: trust, distrust, and uncertain. If obtained trust value is low, CSPj trust value maps to distrust. If obtained trust value is high, CSPj trust value maps to trust and otherwise trust value maps to uncertain.

4. Simulation Results

In this section, the results of two components of cloud trust management frameworks are explained. One component is the feedback evaluation component, and another is the SLA negotiation component.

4.1. SLA Negotiation Component Results. In this section, the simulation results of the demand trust evaluation agent are illustrated. As mentioned in Section 4, the trust evaluation agent is one of the most important agents of the **SLA** negotiation component. In this section, the simulation results of DTEA are described especially.

The task of DTEA is described in Section 4. Simulation parameters are described in Table 1. Also, the batch size is 10, and the rate of entry request follows the Poisson process. The simulation results are shown in Figures 7 to 9.

User service satisfaction with DTEA and without DTEA is shown in Figure 7. In Figure 7, the horizontal axis is arrival time and the vertical axis is service satisfaction. Using DTEA enhances user service satisfaction. Also, using DTEA increases trust values shown in Figure 8. In Figure 8, the horizontal axis is arrival time and the vertical axis is trust value. Due to load balance, DTEA uses a roulette wheel mechanism on resource nodes. Figure 9 illustrates service satisfaction in DTEA with a roulette wheel and without a roulette wheel. In Figure 9, the horizontal axis is arrival time and the vertical axis is service satisfaction. Rolette wheel has a small effect on service satisfaction. It should be mentioned that DTEA with a roulette wheel not only decreases service satisfaction but also makes good load balance on resource nods.

As shown in Figures 7–9, using the demand trust evaluation component and assigning the nearest cloud service provider to the request of the cloud service users, the satisfaction of cloud service users from receiving the service has increased, which is indicated in Figure 7. And, since the increase in satisfaction is directly related to the increase in the degree of trust of cloud service providers, this leads to higher trust values as well as the average values of trust in two cases using DTEA and without using it as shown in Figure 8. As shown in Figures 7 and 8, both cloud service user satisfaction and the trust value of cloud service providers have increased by about 0.2. As stated in the description of the proposed framework, since the use of DTE may lead all requests to a number of cloud service providers and create a bottleneck because the balance of load between cloud service providers is used, the revolving mechanism in demand trust evaluation is used, which Figure 9 shows the percentage of satisfaction of the audience using/without using the revolving mechanism. And as shown in Figure 9, the use of this mechanism of about 0.9 has resulted in higher audience satisfaction from the received services.

4.2. Evaluation of Simulation Result. In this research, a ranked data set called epinion has been used to simulate the component of evaluating the feedback received. This data set is selected from mass and Avesani and contains 6194 items from CSP, 55197 items from CSU, and 394691 items from feedback trust values. The epinion dataset has two modes (trust and distrust). The trust mode is denoted by 1, and the distrust mode is denoted by minus 1. Based on the explanations provided in Section 3, the feedback evaluation

Table 1: DTEA simulation parameters assumption.

Parameters	Description	Numbers
N	The number of a user request for a separate domain	1000
M	The number of resource nodes	10
PN	The number of QOS parameter	4

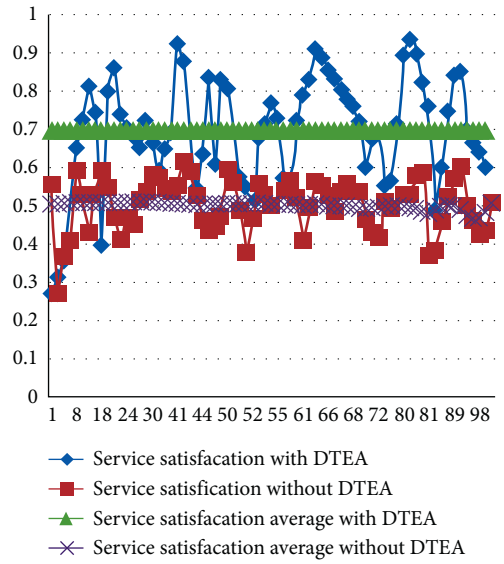

Figure 7: Service satisfaction with DTEA and without DTEA.

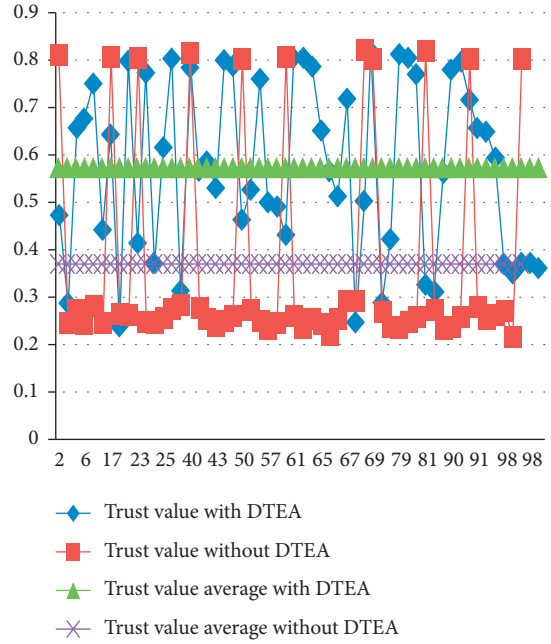

Figure 8: Trust value with DTEA and without DTEA.

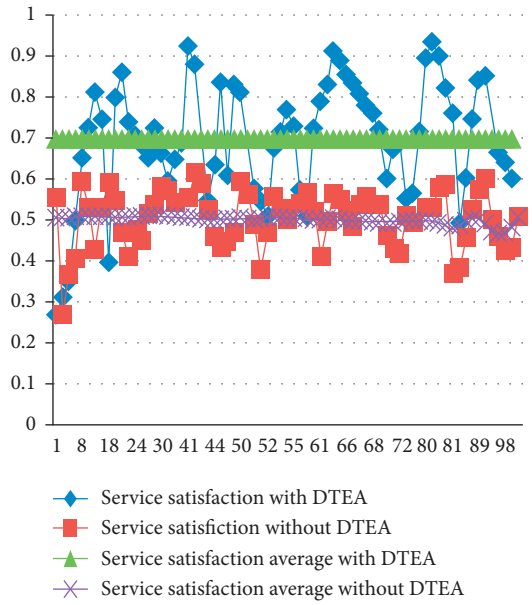

Figure 9: Service satisfaction in DTEA with a roulette wheel and without roulette wheel.

component is able to detect fake feedback. In the simulation performed in this study, we injected fake feedback into the epinion data set. A random injection rate of 20% was considered.

The degree of trust in the fake feedback is corrected using the feedback retrieval component. Figure 10 shows the average amount of feedback trust in three modes (including preinjection, postinjection, and postcorrection). In this figure, the horizontal axis represents the CSP and the vertical axis represents the average values of trust in feedback. To obtain a true approximation to the values of feedback trust, we examine an average of 1000 CSP.

The reliability of feedback based on every 1000 CSPs is shown in Table 2. Based on the results in this table and Figure 10, it can be argued that the average amount of trust in the feedback sent by client users after injection with the amount of confidence in the initial data sets is up to 41% differences. Therefore, by using the feedback component, the proposed method is used to identify and make correction in fake feedback. In this way, the percentage of confidence is reduced to 20%. It can be concluded that the use of this new component increases the efficiency of the trust management framework and ultimately reduces the impact of attacks by malicious users on the trust percentage of feedback received.

As we can see in Figure 10, the average confidence and trust level are greatly reduced (approximately 29%) by injecting fake feedback without the feedback evaluation component. However, after using the feedback evaluation component, the average value of feedback trust increases by 10% and its value is closer to the initial data.

4.3. The Trust Values in Multicloud Computing. In this section, the simulation results of trust values in multicloud computing are presented. Figure 11 shows trust values for three states 2TSPs, 3TSPs, and 4TSPs. In Figure 11, the horizon axis represents the time and perpendicular axis indicating the average trust values, which indicates the conformity of the proposed framework with multicloud environments; for example, the simulation results for 2TSPs, 3TSPs, and 4TSPs are presented.

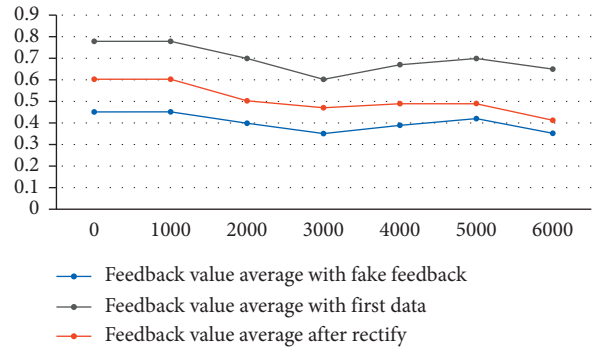

Figure 10: Feedback trust value average.

Table 2: Feedback trust value average (FTVA) based on 1000 CSP.

	1000	2000	3000	4000	5000	6000
FTVA before rectifying with fake feedback	0.45	0.42	0.37	0.38	0.41	0.38
FTVA after rectifying with fake feedback	0.58	0.52	0.47	0.47	0.47	0.39
FTVA without fake feedback (first data)	0.76	0.7	0.61	0.67	0.71	0.65

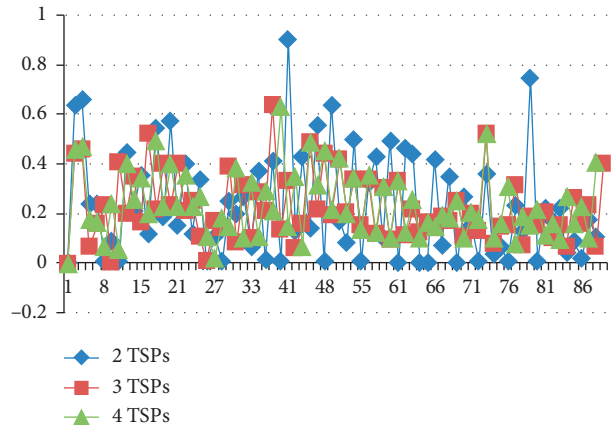

Figure 11: Trust values in multiclouds.

4.4. Comparing Proposed Multiclouds Trust Management Framework with Other Frameworks. In this section, the simulation results of four models (feedback-based model, SLA-based model, multicloud model, and new proposed model) are presented in Figures 12 and 13. Trust is one of the most important parameters that can be used to compare models with each other in the field of trust management. The higher the average trust values, i.e., the framework or method proposed has performed better, the higher the satisfaction of cloud service users from receiving the service, which means that a suitable cloud service provider is

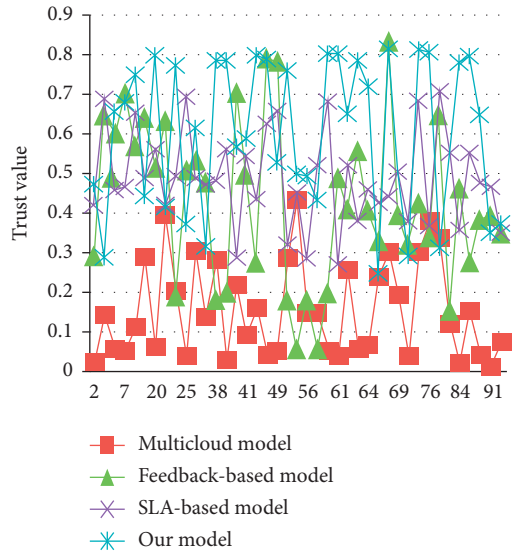

Figure 12: Comparison of the proposed model with other models.

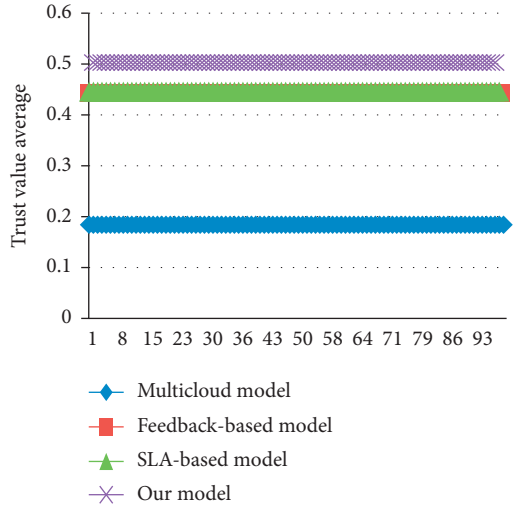

Figure 13: The comparison of trust values means between the new proposed model and other models.

assigned to their request. Therefore, in this study, trust values, as well as average trust values in the same conditions and the same dataset, were used to evaluate the proposed framework and compare it with other methods and models. In the proposed framework, because the assignment of the nearest cloud service provider to the request of the cloud service user is done using the request trust evaluation components, this will lead to higher cloud service user satisfaction and a consequently higher level of trust of cloud service providers. The number of articles and studies that have provided a trust management framework in multicloud environments is limited. The most complete and closest model that is presented in multicloud environments for trust management is the multicloud model that the proposed framework presented in this study has advantages such as detection fake feedback compared with that in the results section of the proposed framework.

Figure 12 shows the comparison of the proposed model with other models. Also, in Figure 13, the comparison of trust mean values between the new proposed model and other models has been presented. In Figures 12 and 13, horizontal axis is time and the vertical axis is the trust values.

As shown in Figures 12 and 13, the proposed framework increased trust values rather than other models. Also compared with other models, our framework gives better mean trust values.

It can be concluded that this new framework is suitable for multicloud environments and gives reliable trust values. Moreover, it is clear that the nearest trust values of other models with the proposed model are obtained in the SLA-based model.

As you can see in Figure 12, the average value of trust in the proposed model is higher than that in the others because in the proposed model, we have a new component named as "demand trust evaluation component". This component selects the service provider which closely matches with other components. The request of CSUs helps a lot, and also the presence of components such as fake feedback detection also helps in having a higher average value of trust than other models, which has not been addressed by a model so far and is one of the advantages of the proposed model.

As can be seen in Figures 12 and 13, in the proposed method, due to the use of the fake feedback evaluation component and also because of the use of the demand evaluation trust component, the mean trust values are at a higher level than the other three models, which indicates better performance of the proposed framework than other frameworks.

The proposed model in this study is compared and simulated with three other models because of the number of proposed models in the field of trust management in multicloud environments that look at trust values as a multidimensional parameter and not a limited one. The proposed model is compared with the three models that have been presented in this field and have such features.

5. Conclusion

In this paper, a new trust management framework for multicloud environments has been proposed.

The advantages of this framework are as follows:

(i) Subjective trust value and objective trust value applied to calculate trust values.

(ii) Objective trust value and subjective trust value are multidimensional parameters.

(iii) Feedback evaluation component was applied in this framework. The Feedback evaluation component task is identifying and rectifying fake feedbacks that any framework does not apply to this component yet. Simulation results had shown the performance of this component, and it shows the effect of this component on trust values.

(iv) Trust negotiation component has used the platform that the output of its component is SLA contract.

One agent of the SLA negotiation component is the demand trust evaluation component. This component selects the CSPs that have the nearest adoption with CSU request, and finally this component causes increase in the service satisfaction and trust values average. The simulation results confirm it.

(v) The proposed framework increased trust values rather than other models (SLA-based model, feedback-based model, and multicloud model). As future work, the trust management model can be proposed along with the detection of fake feedback in other applications such as fog computing and the Internet of Things.

In the case of failures, it can be noted that if several cloud service users colluded with each other and attacked a cloud service provider for a period of time, the proposed feedback evaluation component cannot detect fake feedback from other feedbacks.

As a future work, it is planned to introduce a trust management model with the feature of detecting fake feedback in IoT networks and fog computing. Game theory can also be used to detect fake feedback in the feedback evaluation component of trust management models.

Data Availability

Data are available on request through contacting with safieh.siadat@gmail.com.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] N. Jafari Navimipour, A. M. Rahmani, A. Habibizad Navin, and M. Hosseinzadeh, "Expert cloud: a cloud-based framework to share the knowledge and skills of human resources," *Computers in Human Behavior*, vol. 46, pp. 57–74, 2015.

[2] R. Buyya, C. S. Yeo, and S. Venugopal, "Market-oriented cloud computing: vision, hype, and reality for delivering it services as computing utilities," in *Proceedings of the 2008 10th IEEE International Conference on High Performance Computing and Communications*, Dalian, China, September 2008.

[3] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[4] N. Jafari Navimipour, A. Masoud Rahmani, A. Habibizad Navin, and M. Hosseinzadeh, "Resource discovery mechanisms in grid systems: a survey," *Journal of Network and Computer Applications*, vol. 41, pp. 389–410, 2014.

[5] R. Buyya, R. Ranjan, and R. N. Calheiros, "Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services," in *Proceedings of the International Conference on Algorithms and Architectures for Parallel Processing*, Busan, Korea, May 2010.

[6] P. Xiao, Z.-G. Hu, and Y.-P. Zhang, "An energy-aware heuristic scheduling for data-intensive workflows in virtualized datacenters," *Journal of Computer Science and Technology*, vol. 28, no. 6, pp. 948–961, 2013.

[7] T. H. Noor and Q. Z. Sheng, "Trust as a service: a framework for trust management in cloud environments," in *Proceedings of the International Conference on Web Information Systems Engineering*, Sydney, NSW, Australia, October 2011.

[8] S. Pearson and A. Benameur, "Privacy, security and trust issues arising from cloud computing," in *Proceedings of the 2010 IEEE Second International Conference on Cloud Computing Technology and Science*, Indianapolis, IN, USA, November 2010.

[9] D. A. B. Fernandes, L. F. B. Soares, J. V. Gomes, M. M. Freire, and P. R. M. Inácio, "Security issues in cloud environments: a survey," *International Journal of Information Security*, vol. 13, no. 2, pp. 113–170, 2014.

[10] I. M. Abbadi, "A framework for establishing trust in Cloud provenance," *International Journal of Information Security*, vol. 12, no. 2, pp. 111–128, 2013.

[11] C. Tang and J. Liu, "Selecting a trusted cloud service provider for your SaaS program," *Computers & Security*, vol. 50, pp. 60–73, 2015.

[12] I. M. Abbadi and A. Martin, "Trust in the cloud," *Information Security Technical Report*, vol. 16, no. 3-4, pp. 108–114, 2011.

[13] I. U. Haq, I. Brandic, and E. Schikuta, "Sla validation in layered cloud infrastructures," in *Proceedings of the International Workshop on Grid Economics and Business Models*, Ischia, Italy, August 2010.

[14] W. Conner, "A trust management framework for service-oriented environments," in *Proceedings of the 18th International Conference on World Wide Web*, Madrid, Spain, April 2009.

[15] Z. Malik and A. Bouguettaya, "Rateweb: reputation assessment for trust establishment among web services," *The VLDB Journal*, vol. 18, no. 4, pp. 885–911, 2009.

[16] W. Fan, S. Yang, and J. Pei, "A novel two-stage model for cloud service trustworthiness evaluation," *Expert Systems*, vol. 31, no. 2, pp. 136–153, 2014.

[17] J. Y.-j. Hsu, K.-J. Lin, T.-H. Chang, C.-j. Ho, H.-S. Huang, and W.-r. Jih, "Parameter learning of personalized trust models in broker-based distributed trust management," *Information Systems Frontiers*, vol. 8, no. 4, pp. 321–333, 2006.

[18] C. Uikey and D. Bhilare, "A broker based trust model for cloud computing environment," *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 11, pp. 247–252, 2013.

[19] S. Chakraborty and K. Roy, "An SLA-based framework for estimating trustworthiness of a cloud," in *Proceedings of the 2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, Liverpool, UK, June 2012.

[20] W. Fan and H. Perros, "A novel trust management framework for multi-cloud environments based on trust service providers," *Knowledge-Based Systems*, vol. 70, pp. 392–406, 2014.

[21] S. Song, K. Hwang, and M. Macwan, "Fuzzy trust integration for security enforcement in grid computing," in *Proceedings of the IFIP International Conference on Network and Parallel Computing*, Wuhan, China, October 2004.

[22] H. Liao, Q. Wang, and G. Li, "A fuzzy logic-based trust model in grid," in *Proceedings of the 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, Wuhan, China, April 2009.

[23] S. Siadat, A. M. Rahmani, and M. Mohsenzadeh, "Proposed Platform for improving grid security by trust management system," 2009, https://arxiv.org/abs/0911.0498.

[24] S. Siadat, A. M. Rahmani, and H. Navid, "Identifying fake feedback in cloud trust management systems using feedback

evaluation component and Bayesian game model," *The Journal of Supercomputing*, vol. 73, no. 6, pp. 2682–2704, 2017.

[25] D. Chen, G. Chang, D. Sun, J. Li, J. Jia, and X. Wang, "TRM-IoT: a trust management model based on fuzzy reputation for internet of things," *Computer Science and Information Systems*, vol. 8, no. 4, pp. 1207–1228, 2011.

[26] J. Guo, I.-R. Chen, and J. J. P. Tsai, "A survey of trust computation models for service management in internet of things systems," *Computer Communications*, vol. 97, pp. 1–14, 2017.

[27] I. U. Din, "Trust management techniques for the Internet of Things: a survey," *IEEE Access*, vol. 7, pp. 29763–29787, 2018.

[28] W. Yuan, D. Guan, Y.-K. Lee, S. Lee, and S. J. Hur, "Improved trust-aware recommender system using small-worldness of trust networks," *Knowledge-Based Systems*, vol. 23, no. 3, pp. 232–238, 2010.

[29] C. Ngo, Y. Demchenko, and C. De Laat, "Toward a dynamic trust establishment approach for multi-provider intercloud environment," in *Proceedings of the 4th IEEE International Conference on Cloud Computing Technology and Science*, Taipei, Taiwan, December 2012.

[30] A. K. Sangaiah, A. S. Rostami, A. A. R. Hosseinabadi et al., "Energy-aware geographic routing for real time workforce monitoring in industrial informatics," *IEEE Internet of Things Journal*, 2021.

[31] A. K. Sangaiah, D. V. Medhane, T. Han, M. S. Hossain, and G. Muhammad, "Enforcing position-based confidentiality with machine learning paradigm through mobile edge computing in real-time industrial informatics," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 7, pp. 4189–4196, 2019.

[32] A. K. Sangaiah, D. V. Medhane, G.-B. Bian, A. Ghoneim, M. Alrashoud, and M. S. Hossain, "Energy-aware green adversary model for cyberphysical security in industrial system," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3322–3329, 2020.

[33] H. M. Mousa and G. F. Elhady, "Trust model development for cloud environment using fuzzy mamdani and simulators," *Journal: Internation Journal of Computers and Technology*, vol. 13, p. 11, 2014.

[34] M. J. Sule, M. Li, G. Taylor, and C. Onime, "Fuzzy logic approach to modelling trust in cloud computing," *IET Cyber-Physical Systems: Theory & Applications*, vol. 2, no. 2, pp. 84–89, 2017.

[35] W. Fan and H. Perros, "A novel trust management framework for multi-cloud environments based on trust service providers," *Knowledge-Based Systems*, vol. 70, pp. 392–406, 2014.

[36] S. Kumar, S. Mittal, and M. Singh, "Fuzzy based trust management system for cloud environment," *Advances in Science and Technology Research Journal*, vol. 10, no. 30, pp. 32–37, 2016.

[37] M. Nielsen, K. Krukow, and V. Sassone, "A Bayesian model for event-based trust," *Electronic Notes in Theoretical Computer Science*, vol. 172, pp. 499–521, 2007.

[38] S. M. Habib, S. Ries, and M. Muhlhauser, "Towards a trust management system for cloud computing," in *Proceedings of the 2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, Washington, DC, USA, November 2011.

[39] A. Lakshminarayanan, "Can CRLs provide bandwidth-efficient online certificate status?" in *Proceedings of the 2006 31st IEEE Conference on Local Computer Networks*, Tampa, FL, USA, November 2006.