electronics

MDPI

Article

A Multi-Dimensional and Multi-Factor Trust Computation Framework for Cloud Services

Aisha Kanwal Junejo [1,†], Imran Ali Jokhio [2,*,†] and Tony Jan [3,†]

[1] School of Computing and Mathematics, Keele University, Newcastle ST5 5AX, UK; a.junejo@keele.ac.uk
[2] School of Information Technology and Engineering, Melbourne Institute of Technology, Melbourne, VIC 3000, Australia
[3] Centre for Artificial Intelligence Research and Optimization, Design and Creative Technology Vertical, Torrens University, Sydney, NSW 2007, Australia; tony.jan@torrens.edu.au
* Correspondence: ijokhio@mit.edu.au; Tel.: +61-410917952
† These authors contributed equally to this work.

Abstract: In this paper, we propose a novel trust computation framework (TCF) for cloud services. Trust is computed by taking into consideration multi-dimensional quality of service (QoS) evidence and user feedback. Feedback provides ample evidence regarding the quality of experience (QoE) of cloud service users. However, in some cases, users may behave maliciously and report false feedback. Users can carry out collusion and Sybil attacks to slander/self-promote cloud services. Trust computed in such cases could be misleading and inaccurate. Evaluating the credibility of user feedback can help in not only preventing the collusion and Sybil attacks but also remunerating the affected cloud services. Despite the advantages of credibility evaluation, very few studies take into consideration feedback credibility and multi-dimensional evaluation criteria. Considering the limitations of existing studies, we propose a new TCF in which trust is computed by aggregating multi-dimensional evidence from QoS and QoE. We have used multi-dimensional QoS attributes to compute the objective trust of cloud services. The QoS attributes are divided into three dimensions, i.e., node profile, average resource consumption, and performance. The node profile of a cloud service is attributed to CPU frequency, memory size, and hard disk capacity. The average resource consumption is quantified based on the current CPU utilisation rate, current memory utilisation rate, current hard disk utilisation rate, and energy consumption. Moreover, the performance of a cloud service is measured by the average response time and task success ratio. Besides that, the credibility of feedback is evaluated to prevent the malicious behaviour of cloud users. Our results demonstrate the effectiveness of our proposed TCF in computing accurate trust in cloud services.

Keywords: cloud services; trust; objective trust; subjective trust; user feedback; regression; credibility; Sybil; collusion

check for updates

Citation: Junejo, A.K.; Jokhio, I.A.; Jan, T. A Multi-Dimensional and Multi-Factor Trust Computation Framework for Cloud Services. *Electronics* **2022**, *11*, 1932. https://doi.org/10.3390/electronics11131932

Academic Editor: George Angelos Papadopoulos

Received: 24 April 2022
Accepted: 19 June 2022
Published: 21 June 2022

1. Introduction

With the plethora of cloud services in the marketplace, the selection of a trustworthy cloud service is often a difficult task for a consumer. The service offerings from different cloud service providers (CSP) claim to meet the highest quality of service (QoS), confirming service level agreement (SLA), and regularity compliance, namely, General Data Protection Regulation (GDPR) [1] and Safe Harbor [2]. In such cases, how a cloud service user (CSU) can get an assurance that a CSP actually complies with the SLA and takes into consideration the user preferences.

Trust plays an important role in establishing a dependable relationship between a CSP and CSU. To date, numerous trust computation models have been proposed in the literature wherein the trustworthiness of a cloud service is assessed on various dimensions, namely, service quality, user feedback, competence, integrity, benevolence, honesty, capability, etc. [3,4]. However, the existing models have a number of limitations, as discussed below.

First, trust computation is based on a single source of evidence, i.e., QoS, expert opinion, or user feedback. Second, trust computation is carried out by CSPs themselves, which raises concerns over the transparency of the process. The CSPs might not take into consideration the negative user feedback or blacklist such users. Third, in trust computation models based on subjective logic [5–8], the weights are assigned by the experts, which in some cases might not be accurate as human judgements are often prone to error. Fourth, the credibility of user feedback is not evaluated, meaning that in some cases, CSUs may behave maliciously and report false feedback regarding their quality of experience (QoE). CSUs can carry out collusion and Sybil attacks to slander/self-promote cloud services. Therefore, the trust computed in such cases could be misleading and inaccurate.

1.1. Motivation

Considering the limitations of existing approaches, in this paper, we propose a novel trust computation framework (TCF) for cloud services. In our proposed TCF, the trust of a cloud service is computed by fusing evidence from monitored QoS and user feedback. The multi-faceted trust assessment guarantees that several factors are considered in trust computation. As discussed above, the trust in a cloud service is computed by the aggregation of QoS evidence and user feedback. To make it clearer, we introduce two notions of trust, namely, objective and subjective. From here onwards, trust computed from QoS would be referred to as objective trust, whereas the trust computed from user feedback would be referred to as subjective trust.

For the objective trust computation, we employ the random forest regression model. Machine learning models, specifically regression and classification, have already been used for solving various computational problems. Likewise, we believe regression models could be employed to predict the trust in a cloud service based on a number of features related to the service quality. Another advantage of regression models is that weight allocation is automatic and eliminates the need for an expert opinion or manual weight inputs.

Moreover, the trust computation process is entrusted to a trusted third party, namely a broker. The trust appraisal by the broker ensures that the trust process is transparent and preserves the privacy of user feedback. Additionally, our proposed TCF includes several modules for user feedback credibility assessment, which subsequently prevents malicious behaviour, namely collusion and Sybil attacks. Computing the credibility of user feedback can help in not only preventing collusion and Sybil attacks but also adjusting the trust of affected cloud services. Besides that, we have observed that very few studies take into consideration feedback credibility and multi-dimensional feedback criteria for subjective trust. The multi-dimensional QoS and QoE evidences enable a precise and accurate trust assessment, which subsequently helps the CSPs to better tailor and/or personalize the cloud services as per the user's requirements.

1.2. Contribution

Following are our main contributions. First, a novel TCF for precise and accurate trust computation of cloud services is proposed. Second, a feedback credibility assessment model that protects against collusion and Sybil attacks is proposed.

The rest of this paper is organized as follows. The related work is discussed in Section 2. Our proposed TCF is elaborated upon in Section 3. The experimental results of the proposed TCF are presented in Section 4. The conclusion and future work are discussed in Section 5.

2. Background and Related Work

In this section, we will first present a brief overview of the trust models. Next, we will review recent trust models proposed for cloud services. We will also discuss the trust models proposed for fog computing. We will then describe the models based on machine learning as it seems to be a robust approach for predicting trust.

The trust computation models are categorized based on the type of evidence, parameters, and the methodology used for aggregating the evidence. The evidence can be subjective, for example, the feedback of cloud service users, and it can be objective, where the QoS is evaluated by monitoring the various service parameters. The depth and breadth of the literature on trust models is immense, and various studies have based their trust computation on different parameters; for example, some use resource availability, hardware specifications, task success ratio, and other use competence, security, privacy, and trustworthiness. Some studies compute trust solely based on one type of evidence, QoS parameters or user feedback.

When it comes to trust computation models, there are two main types, (1) subjective trust and (2) objective trust. Subjective trust is quantified based on the direct interactions between the trustor and trustee. As mentioned above, subjective trust is computed based on user feedback, reviews, and ratings. The experts in certain fields often use their personal experience to evaluate the interactions between systems and users. Objective trust is computed based on the evidence gathered from multiple sources, similar to a reputation-based approach. The evidence for objective trust computation is gathered by deploying real-time monitoring tools. The evidence can be aggregated in two ways, subjective logic [5] and real-time adaptive trust computation [9,10]. The main difference between the two approaches is the manner in which weights are assigned; in the former, they are assigned manually and/or based on personal preferences, and in the latter, they are assigned using the maximizing deviation method [10], information entropy [9], and regression [11–13]. Besides these, some studies employ both subjective and objective trust to evaluate the trustworthiness of systems [6,10,14,15]. Such a hybrid approach avoids the system/user bias and guarantees that trust is computed based on different types of collective evidence. Therefore, our review of the literature is also based on the aforementioned trust computation approaches, and for each study we cite, we will describe the evidence, parameters, and trust aggregation method, objective, subjective, and hybrid.

Recently, Manel et al. [16] proposed a trust model for cloud service selection. The model is based on a Naive Bayes probabilistic classifier and uses the correlation among QoS parameters to predict missing ratings. This work only used objective evidence and do not consider the user feedback in service selection. The QoS evidence alone is not enough to assess the trustworthiness of cloud services, as it can be fabricated by the providers and necessitate employing user's QoE ratings in trust computation. Balcao et al. [17] propose a trust assessment framework for cloud providers. This is an interesting work that employs multi-dimensional evidence about governance, transparency, information security, and QoS parameters to compute the trust. It also considers the user feedback but does not evaluate its credibility before incorporating it into the trust model. Another limitation of this work is rather simplistic formal modelling based on arithmetic and geometric means. Such models cannot capture the recentness of the evidence and will use all available data in the computation, and can produce inaccurate results regarding the performance and trustworthiness of CSPs. In addition to that, it is not clear how the consumers can rate crude qualitative dimensions of governance, transparency, and security, as usually naive users do not have access to tools that are required to assess these. Ahmed et al. [18] propose an objective trust computation model for the selection of dependable service from a federation of cloud services. The assessment is based on QoS parameters consisting of node profile, reliability, and competency. This model neither considers user feedback nor provides any attack resiliency.

Golnaz et al. [19] propose a subjective trust model that employs user feedback to assess the trustworthiness of CSPs. In comparison to other studies [20,21] that also incorporate feedback, this work proposes some components that can improve the quality of user feedback by addressing the data sparsity and invalidity problems. A key limitation of this study is the inconsideration of objective evidence. Doa et al. [22] conducts a study to compare existing trust models, analytical hierarchy process (AHP), fuzzy analytical hierarchy process (FAHP), fuzzy technique for order preference by similarity to ideal solution (FTOPSIS), multi-criteria decision-making (MCDM), and the technique for order preference by similarity to ideal solution (TOPSIS). These models are compared via ratings gathered on performance, agility, finance, security, and usability criteria using a hybrid fuzzy logic method and multi-criteria decision-making techniques. The comparison is based on QoS evidence gathered from the Cloud Analyst application, which is part of the widely used CloudSim simulator. This study also does not take into consideration user feedback. Chen et al. [23] propose a hierarchical trust model for IoT services. The consumers report their first-hand experience and recommendations, which are later used in computing subjective trust by incorporating the social relationships between IoT devices. This model provides some security against malicious attackers. Compared to other studies, this work does not consider the objective evidence in trust computation.

Among several parameters that are used to compute trust, competence and trustworthiness are the most important ones. The work in [15] approximates the risk of interactions with new and/or unknown cloud service providers based on these two parameters. Trustworthiness is evaluated on the basis of the direct interactions with the vendors and their established reputation. In other words, it is a subjective evaluation. For competence, the service level agreements (SLAs) are assessed for transparency in outlining the quality of service (QoS) parameters and the relevant terms and conditions. The study in [24] proposes a trust computation model that uses both subjective and objective evidence to compute the trust for cloud services. The work in [6] devises a hybrid model to evaluate the trustworthiness of cloud services. The objective trust is quantified based on the real-time parameter measurements, whereas the subjective trust is quantified on the basis of prior direct experiences and the recommendations of others if first-hand evidence cannot be acquired. The study in [25] devised a service ranking algorithm for helping the system designers and users to select highly rated cloud services.

The work in [7] uses only one source of objective evidence to compute the trust based on subjective logic. It completely disregards user feedback. In contrast to the above study, the work in [8] evaluates the trustworthiness of the web services based on user preferences. The trust is computed using subjective logic and considers the credibility of the user ratings as sometimes malicious users can give false feedback to decrease/increase the trust. A major limitation of this work is the inconsideration of QoS parameters, making it unsuitable for cloud services. Another work that evaluates the credibility of user feedback feeding it into computation, is proposed in [21]. The reputation-based trust model is based on the subjective trust that is computed by taking feedback on multiple dimensions, namely, usability, accessibility, availability, price, customer service, etc. Additionally, a feedback credibility model is proposed to protect cloud services against malicious behaviour (e.g., collusion or Sybil attacks) from its users. The collusion attacks are detected by quantifying the number of feedbacks given by a CSU and subsequently lowering the impact of multiple feedback ratings by the feedback density parameter. This approach is not a viable solution for real-world cloud services, as only a certain amount of feedback can be considered in a trust computation while the rest is discarded. Besides that, a major limitation of this work [21] is that trust is computed from user feedback only. However, it is evident that both direct evidence and user feedback should be considered in assessing the trustworthiness of cloud services. The real-time evaluation of QoS parameters can give actual insights into service quality.

Next, we discuss trust models that use fog computing as the trust computation entity. The study in [26] proposes a trust model for vehicular ad hoc networks (VANETS). The dependability and reliability of such systems highly rely on data integrity such that messages are not modified when they are transmitted from one vehicle to another. This work uses experience and plausibility as trust parameters to compute the trust using fuzzy logic. Evaluating the trustworthiness of entities in multi-hop sensor networks is essential to guarantee the dependability of the system. The work in [27] designs a hierarchical trust model for evaluating the performance of the sensor devices, edge devices, and service providers. The trust is computed at the fog layer to ensure the trust ratings are accessible in real-time with no and/or negligible delay and latency. Recently, the work in [13] proposes a trust management system to compute the trust for different entities in a fog-based cyber-physical system (CPS). This is an objective trust computation model wherein the final trust in fog nodes is computed by combining the evidence gathered from IoT devices and a fog monitoring node. The credibility of evidence coming from CPS devices is evaluated to guarantee the inputs are secure and have not been modified by malicious adversaries. Next, we discuss the machine learning-based trust models.

Since machine learning is widely being used for classification and prediction problems, some the recent studies have employed it to predict the trust of cloud services based on a number of features. The work in [12] uses multiple linear regression (MLR) to predict the trust of sensor networks wherein the cloud provides the backend services. Likewise, in [11], logistic regression is used to compute the trust in MANETS.

Discussion

Now we draw on some conclusions from our review of the related studies. Table 1 presents a comparison of state-of-the-art trust models for cloud services. The **Type** column defines the type of trust model, the **Evidence** column lists the evidence used, the **Aggregation Method** column describes the method used to combine/fuse the evidence, and the **Security Measures** column describes the countermeasures against attacks. It is believed that a multi-dimensional assessment of cloud services is essential to achieve better insights into QoS and QoE. It can be observed that, except ours, only two more studies [17,24] use objective and subjective evidence for evaluating the trustworthiness of cloud services.

Table 1. State-of-the-Art Trust Models.

Study	Type	Evidence	Aggregation Method	Security Measures
[7]	Objective	QoS Parameters	Subjective Logic	×
[8]	Subjective	User Preferences	Weighted Sum	×
[16]	Objective	QoS	Naive Bayes and n-gram Markov model	×
[18]	Objective	Node Profile, Reliability, and Competency	Weighted Sum and Utility Function	×
[17]	Objective and Subjective	Governance, Transparency and Security	Arithmetic Mean Geometric Mean	×
[19]	Subjective	User Feedback	Beta Reputation	Sparse Feedback and Credibility Evaluation
[23]	Subjective	User Ratings	Weighted Sum and Cosine similarity	×
[15]	Subjective	Competence and Trustworthiness	Belief Theory and Dempster–Shafer Theory	×
[24]	Objective and Subjective	QoS Parameters	Objective Entropy, Trust Preference, Weighted Sum	×
[21]	Subjective	User Feedback	Weighted Sum	Credibility Evaluation
[27]	Objective	Response Time, Packet Delivery, and Energy Consumption	Multiple Linear Regression	Data Attacks
[11]	Objective	Energy, Capability, and Profit	Logistic Regression	Attack Resilience
Ours	Objective and Subjective	QoS and User Feedback	Weighted Sum Random Forest Regression	User Feedback Credibility

Incorporating user feedback in trust computation provides insights into the service quality experienced by CSUs, but, at the same time, it also provides an opportunity to adversaries to exploit it to change (increase/decrease) the trust of CSPs. The identification of malicious feedback is vital to prevent malicious behaviour (i.e., Sybil and collusion attacks) of CSUs and subsequently computing a precise and accurate trust of cloud services. From Table 1, only five studies, including ours [11,19,21,27], have some mechanism to address the security issues in user feedback. Interestingly, all of these studies are either based on objective or subjective assessment, and none of them fuses multiple sources of evidence like this study. Furthermore, adaptive trust evaluation approaches are better than subjective logic as they do not require weight inputs from experts. It is better to adopt a data-driven approach to compute the trust and derive the weights based on the features gathered from objective evidence, QoS parameters in this case. Machine learning algorithms such as multiple linear regression and random forest regression can be used for trust prediction.

Considering the limitations of existing studies, we propose a new TCF in which we evaluate cloud services on multi-dimensional criteria and apply a credibility model to countermeasure the malicious behaviour of cloud users. We predict the objective trust of cloud services by using a random forest regression model.

3. Proposed Trust Computation Framework

Now, we introduce our proposed framework that enables the broker to precisely and accurately compute the trust of cloud services. Figure 1 illustrates the proposed TCF. There are three types of entities, namely, cloud services, CSUs, and a broker. Infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) are three popular cloud services. Moreover, nowadays, some special and/or dedicated services (i.e., security, trust) are also offered by CSPs. Since the cloud services are offered by CSPs, the trust of a service reflects the QoS provided by the CSP. From here onwards, the terms cloud service and CSP are used interchangeably. Brokers assist CSUs in finding services matching their requirements, which are offered by various CSPs. Moreover, brokers are also responsible for trust computation and management. Each CSP is required to register its services with the broker.

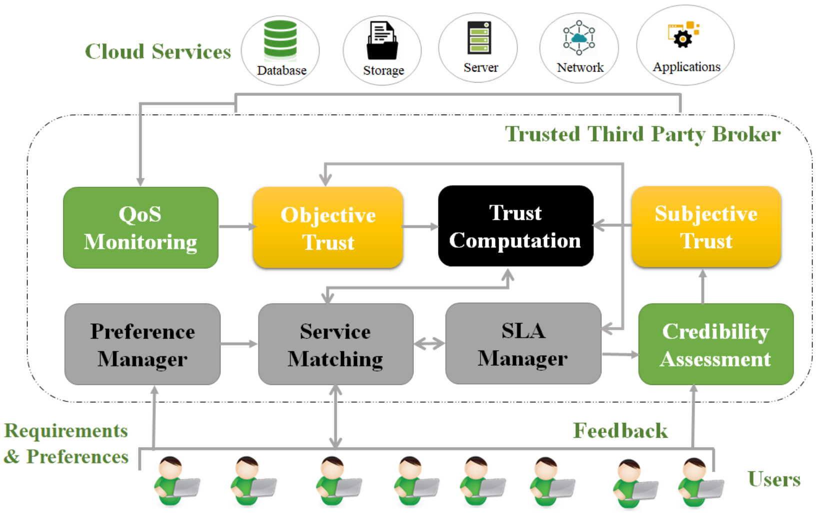


Figure 1. Proposed Trust Computation Framework.

Trust gained by a cloud service reflects its performance based on directly monitored QoS evidence and user feedback. To be more specific, the trust score of a cloud service is the aggregation of trust computed from the evidence gathered by the broker and CSUs. The broker monitors the QoS parameters and computes the objective trust. The CSUs also provide feedback regarding their experience with the CSP. The feedback is subsequently sent to the broker, which computes subjective trust on behalf of CSUs. The various QoS parameters and feedback factors incorporated in trust computation are listed in Table 2. It is underlined that for all types of trust, namely, objective, subjective, and final trust of cloud services, trust lies in the range of $[0, 1]$. It is represented as a fraction. The proposed TCF consists of eight submodules, namely:

1. Preference Manager
2. Service Matching
3. SLA Manager
4. Credibility Assessment
5. QoS Monitoring
6. Objective Trust
7. Subjective Trust
8. Trust Computation

The details of each of these modules are given below.

Table 2. Cloud Service Trust Parameters.

Broker QoS Parameters [13]	Cloud User Feedback Parameters
CPU Frequency Memory size Hard disk capacity Current CPU utilisation rate Current memory utilisation rate Current hard disk utilisation rate Energy Consumption Average response time Average task success ratio	Availability Accessibility Response time Customer service Price Speed Technical support Storage space Features Ease of use

3.1. Preference Manager

This module handles the user preferences on the basis of their service requirements. Service specifications, i.e., processing, computational, and storage, for an IaaS can be defined in the form of requirements, whereas the security, privacy, and legal aspects can be covered in preferences.

3.2. Service Matching

Initially, the CSPs register all their services with the broker. The service matching module takes service requirements as an input from the preference manager and performs the resource matching on the available cloud services. The resource matching is performed on the basis of a well-defined criterion. For service selection, the broker checks the trust of services that have matched the user requirements and selects the one with the highest trust value.

3.3. SLA Manager

Once the CSU selects a particular cloud service, it will negotiate the SLA details with the CSP provider. An SLA contract is signed between the CSU and CSP based on a number of required service quality attributes. The SLA guarantees a minimum expected level of service quality between the CSUs and CSPs.

3.4. QoS Monitoring

This module monitors the QoS parameters of CSP by pulling in the run-time service statistics. Our choice of parameters is motivated by [13], but they have used it for fog nodes. Since the cloud services are like large-scale fog nodes, it was rational for us to use the same parameters as them. The broker monitors four kinds of parameters (see Table 2), namely (1) service specification, (2) average resource usage, (3) average response time, and

(4) average task success ratio. The service specification profile includes CPU frequency, memory size, hard disk capacity, and energy consumption. The average resource usage information consists of the current CPU utilisation rate, current memory utilisation rate, current hard disk utilisation rate and current bandwidth utilisation rate. The broker stores the QoS data in a monitoring database and later uses it for objective trust computation.

3.5. Objective Trust Computation

Once the service quality is monitored by the broker, the next task is the objective trust computation. For $\mathcal{T}_{obj}$, the broker acquires/quantifies eight different QoS parameters listed in Table 2, at discrete time instances $t = \{1, 2, \ldots, i, \ldots, j\}$. For evaluating the service quality, the objective trust prediction is modelled as a random forest regression problem. The monitored QoS parameters (check Table 2) are the input features.

$\mathcal{T}_{obj}$ at time $t = i$ is predicted using a random forest regression model established using available training samples that include input feature sets and labelled output of $\mathcal{T}_{obj}$. The random forest regression is a type of supervised learning model that learns from the training sample data (including ground truths in the output samples). The random forest regression model makes predictions by combining the predictive outputs of several decision tree models.

Let us suppose if there are n number of decision tree models, each denoted by r_m, then the regression model is formalized using Equation (1):

$$r(x) = \sum_{m=1}^{n} \alpha_m r_m(x) \tag{1}$$

where α_m is the weight assigned to each decision tree (with $\sum\limits_{m=1}^{n} \alpha_m = 1$ and $\alpha_m \in [0, 1]$) and x is the input data. r is the weighted sum of base model outputs (i.e., equal to the mean if the weights are 1/m for all decision trees). Each decision tree is trained on a subsample of the data.

The next task is to select the best feature from the high-dimensional QoS features to split the data. The first step in this process is optimisation, wherein a single feature is randomly selected and optimized at each node in a decision tree. The above process is repeated until all features have been optimized. Once the best feature is found from the previous step, the dataset is split and passed to the two child nodes in the decision tree.

If $S = \{s_1, s_2, \ldots, s_p, \ldots, s_q\}$, it denotes q trusted cloud services. $\check{o}_t(s_p)$ denotes the trust level of a cloud service, s_p, at a given time t. $\check{o}_t(s_p)$ is predicted based on the QoS measurements pulled by a broker at that time instance. The objective trust $\mathcal{T}_{obj}$ of $\check{o}_t(s_p)$ over a time period (i.e., one hour or one day) is computed using Equation (2):

$$\mathcal{T}_{obj} = \sum_{t=1}^{i} (\check{o}_t(s_p) \times \omega_t(s_p)), \tag{2}$$

where $\omega_t(s_p)$ is the weight assigned to each $\check{o}_t(s_p)$ with $\sum\limits_{t=1}^{j} \omega_t(s_p) = 1$ and $\omega_t(s_p) \in [0, 1]$. ω is a time-decay-based weight computation function, and gives more weight to recent interactions, as shown in Equation (3).

$$\omega_t = \frac{(1 - (t+1)^{-1})}{\sum\limits_{t=1}^{i} (1 - (t+1)^{-1})} \tag{3}$$

3.6. Subjective Trust Computation

The subjective trust $\mathcal{T}_{sub}$ of a cloud service s_p is computed as follows:

$$\mathcal{T}_{sub} = \frac{\sum_{j=1}^{v} F(s_p, j)}{v} \tag{4}$$

where $F(s_p, j)$ is the feedback for pth cloud service.

3.7. Feedback Credibility Assessment

In our proposed TCF, subjective trust is computed in two steps. In the first step, multi-dimensional feedback (Table 2) is taken from CSUs. While in the second step, the credibility of the feedback is evaluated, and the resulting feedback is employed to compute subjective trust. The user feedback can be subject to various malicious attacks, namely, Sybil, collusion, and on-off. Considering the possible risks, it is essential to verify the credibility of user feedback. Accordingly, a trust credibility evaluation module is designed to prevent and mitigate collusion and Sybil attacks.

3.7.1. Sybil Attacks Prevention

In order to prevent Sybil attacks, all CSUs are required to register with the broker based on a unique set of credentials i.e., username, personal details, address, etc. The CSUs can register after signing the SLA with CSP. Only authenticated CSUs can give feedback. The broker maintains a database in which it stores the hashes of credentials that belong to the CSUs. During the registration process, the credentials are hashed and aggregated.

Let $H(:)$ be a secure cryptographic hash function such as SHA3. Let $C = c_1, c_2, c_3, \ldots, c_m$ be a set of user credentials. The broker calculates $H(c_1)||(c_2)|| \ldots ||(c_m)$ and compares it with the existing user records already stored in the CSU database. If a match is found, the broker will generate a warning that CSU is already registered; otherwise, it will register the new record.

3.7.2. Collusion Attacks

A collusion attack occurs when a user sends multiple false feedbacks to increase/decrease the trust of a cloud service. Such attacks can occur strategically and occasionally. In strategic collusion attacks, a number of compromised CSUs collaborate to report false feedback. While in occasional attacks, the CSUs can periodically change the feedback behaviour. Sometimes they give high feedback, while at other times, they give low feedback. The attackers adopt such different behaviours in order to remain undetected and still be able to change the trust.

The prevention of collusion attacks is not trivial. It might be possible that some CSUs give malicious feedback (i.e., very high and very low) to push the trust closer to either zero or one. For example, if users give very high feedback (i.e., [0.7–1]), it can result in a value closer to one. Likewise, with lower feedback in a range of [0.1–0.3], the trust can decrease to 0. Now the problem is one cannot be certain about the malicious behaviour; at times low and high values can be given by legitimate CSUs who did not experience a good quality of service or otherwise. The users can give good/bad feedback based on their experience. However, a collaborative effort by some CSUs to push trust in one direction or the other can be taken into consideration to adjust the subjective trust in cloud services.

Considering these constraints, we consider time as an important factor for collusion attack prevention. The rate of change of subjective trust is analysed by comparing the trust in consecutive time instances. Moreover, the collaborative effort to change trust is also handled by reducing its impact accordingly. Precisely, the collusion is detected by “subjective trust variance” and “volume collusion” parameters. Next, we discuss each of these parameters in detail.

3.7.3. Subjective Trust Variance

The subjective trust variance λ measures the change in trust during consecutive time instances and subsequently compares it with a predefined tolerance threshold τ. If the difference λ in two time instances is greater than τ, then the subjective trust in the current time instance is adjusted.

$$\lambda = |\mathcal{T}_{sub}^{i-1} - \mathcal{T}_{sub}^{i}| \tag{5}$$

3.7.4. Volume Collusion

This parameter quantifies how many CSUs are collaborating to change the subjective trust in a given time instance. The impact of feedback given by a significant number of collaborating users is further reduced using Equation (6).

$$\Upsilon = \frac{m}{\mathbf{v}} \times \lambda \tag{6}$$

where m is the number of collaborating CSUs that are giving similar kinds of feedback. $\mathbf{v}$ denotes the total number of the feedback given to a cloud service s_i (i.e., feedback volume), and λ is subjective variance. The aggregated subjective trust is considered credible if volume collusion is higher.

3.7.5. Feedback Credibility

The broker improves the accuracy of trust computation by computing the credibility of false feedback using Equation (7). The credibility is computed in two cases. The first case of credibility is based on volume collusion Υ that detects the change in trust due to a significant number of malicious CSUs. In the second case, credibility is computed by considering the subjective variance λ that quantifies the change in trust during consecutive time instances.

$$C_r = \begin{cases} \Upsilon, & \text{if } \lambda > \tau \quad \text{and } m > p, \\ \lambda, & \text{if } \lambda > \tau \end{cases} \tag{7}$$

where p is the minimum percentage of CSUs, which is assumed to affect the subjective trust. Next, the subjective trust in a recent time instance is adjusted using Equation (8).

$$\mathcal{T}_{sub}^{i} = \mathcal{T}_{sub}^{i-1} \pm C_r \mathcal{T}_{sub}^{i} \tag{8}$$

3.8. Trust Computation

Having presented the objective and subjective modules, the next step is to compute the final trust $\mathcal{T}$ using Equation (9).

$$\mathcal{T} = \alpha \times \mathcal{T}_{sub} + (1-\alpha) \times \mathcal{T}_{obj} \tag{9}$$

where α is the weight of $\mathcal{T}_{sub}$, and $(1-\alpha)$ is the weight of $\mathcal{T}_{obj}$. If $\alpha = 0$ in Equation (9), then only objective trust $\mathcal{T}_{obj}$ is used in computing $\mathcal{T}$, and if $\alpha = 1$, then $\mathcal{T}$ is computed based on subjective evidence alone. As can be observed that both $\mathcal{T}_{sub}$ and $\mathcal{T}_{obj}$ play a role in evaluating the trustworthiness of the cloud services; it is better to assign them either equal weights or decide their proportion based on the number and recentness of the evidence. A few studies maintain that $\mathcal{T}_{obj}$ should be assigned a high weight. Since objective evidence indicates the real performance of the cloud services based on the resources they have and how well they completed each task. On the contrary, if the amount of feedback from CSUs is higher, then $\mathcal{T}_{obj}$ should get a high value.

4. Experimental Results and Analysis

In this section, we present the results of our proposed TCF. We begin by stating the objectives of our experiments. Following this, we discuss the testbed and dataset used for computing the trust of cloud services. Next, we present the experiments that were set up to evaluate the proposed framework. Experiment 1 illustrates that trust results in a legitimate environment wherein all users are honest. Experiment 2 demonstrates the effectiveness of adding a credibility assessment module in the subjective trust computation. Experiment 3 presents the comparative analysis results. All these results together demonstrate the advantage of aggregating two sources of evidence to compute trust and also underline how important it is to evaluate the credibility of user feedback as it can impact the trust accuracy if left unverified.

4.1. Experimental Objectives

The first objective of the experimental evaluation is to underline the effectiveness of the multi-factor trust assessment of cloud services. The second objective is to analyse the robustness of the proposed TCF in preventing the malicious behaviour of CSUs. Collusion attacks are designed in different configurations of a hostile strategic environment whereby the CSUs are collaborating to change the trust of cloud services. Moreover, the third objective is to perform a comparative analysis of the final trust $\mathcal{T}$ computed from our TCF with the one computed from an existing study [10], wherein the increasing number of positive feedback from CSUs can increase the overall trust degree. The major limitation of [10] is the inability to prevent attacks on user feedback, namely, Sybil and collusion.

4.2. Testbed

Here, we describe the setup of our experimental testbed. As discussed in Section 3, the broker is responsible for efficient resource matching and trust computation. The broker maintains a database of CSUs and CSPs. In our experimental prototype, the broker is simulated by creating a web-based interface where different CSPs register by specifying the types of services (i.e., IaaS, PaaS, SaaS, etc.) they offer, as specified in Table 3 . At the backend, the broker interface is connected with the Greencloud simulator [28]. The broker takes the service requirements as input and searches the registered CSPs, and subsequently returns the matching resources along with their trust values. The values of different parameters used in our experiments are listed in Table 4.

Table 3. Cloud Services.

CSP ID	Service Type	Service	Duration	Bill
CSP1	IaaS	Virtual Machine	6 months	37 USD
CSP2	PaaS	Linux Platform	9 months	100 USD
CSP3	SaaS	Apps	3 months	200 USD
CSP4	DaaS	My SQL	12 months	150 USD
CSP5	IaaS	Virtual Machine	4 months	300 USD

Table 4. Simulation Parameters.

Parameters	Values
Number of Brokers	1
Number of Cloud Services	113
Number of Cloud Users	7000
Number of Feedback	10,000

4.2.1. Objective Trust Parameters

In order to compute $\mathcal{T}$, the broker requires direct QoS evidence to compute $\mathcal{T}_{obj}$, and user feedback to compute $\mathcal{T}_{sub}$. $\mathcal{T}_{obj}$ is computed by monitoring QoS-based evidence gathered from the Greencloud simulator. Each data centre represents a CSP in our case. User requirements for 113 cloud services (Table 2) are defined and later simulated to measure the

QoS parameters. We measured the CPU frequency, memory, hard disk capacity, andenergy consumption and, later on, computed the current CPU, memory, and disk utilisation. We also quantified the average response time and task success ratio for each cloud service. The user requirements were scaled over time to reflect the increasing load on the cloud services.

In Section 3.5, we discussed that the random forest regression model is used to predict $\mathcal{T}_{obj}$. The random forest regression model is a supervised machine learning algorithm, so it needs labelled outputs for training and testing. However, the Greencloud [28] simulator that we have used to create cloud services and gather their QoS parameters does not give us a trust value. Therefore, our next task was to find a way to generate the trust labels for simulated cloud services based on how well they performed and how many tasks they successfully completed. We have compared the obtained QoS parameters with the values stored in SLA contracts. We have used "task success ratio" as the key feature to compute the $\mathcal{T}_{obj}$ trust value, meaning that it gets a high value if the average task success ratio is high, and the QoS parameter values match with the SLA contract and vice versa. This work followed the well-established standards of labelling the ground truths (in both training and testing data) [13,16,21] based on prior domain information during the data preparation stages. The domain information includes the values of the input feature set compared with the predefined SLA values to check how much they vary from the guaranteed service quality, the verification that the cloud service has successfully completed a given task, and how long it took. In other words, the label $\mathcal{T}_{obj}$ is generated based on the node profile, average resource consumption, and performance features. This experimental analysis used a software program that checked both of these conditions and generated the labels. It is noted that the task success ratio is an important parameter in generating the labels because if the values of other input features are in conformance with the SLA but the cloud service is completing half of its tasks, then it cannot be considered trustworthy. These steps were closely followed according to the benchmarking references. Additionally, this experiment did not use the task success ratio as a feature of the regression as it was already used in generating the labels. Once the ground truths in training and testing data are available, the purpose of the regression or machine learning model is to identify the unknown function/relationship between the input and output within the training samples. Once a regression or machine learning model is established, another dataset is used to test the model's fitness and usefulness.

Train-Test Split: It is underlined that in the random forest regression experiment, 70% of the records are used for training, whereas 30% are used for testing. Next, we partitioned the generated dataset into a train–test split and then rigorously trained the random forest regression model.

Selection of Appropriate Regression Model: As we know, several machine learning models, namely, multiple linear regression, support vector machine, random forest regression, and neural networks, can be used for prediction. We have compared the prediction accuracy of these models and found that the random forest regression model outperforms the others (in consideration of robustness as well). After intensive training of all the regression models, we tested them with the remaining 30% of the data samples. The classification error rate for the neural networks was quite high, and that of the support vector regression was reasonable but showed signs of overfitting. As per our expectations, the random forest regression model correctly classified the trust labels with a 98% accuracy against the ground-truth trust label values available in the testing sample data. Note the testing sample data are different from the training sample data to validate the model. We believe the random forest regression achieved good accuracy and robust modelling due to its non-linear regulated model and, therefore, is well suited for these tasks. The error rate for multiple linear regression on the testing data was a bit high, 11%. The error rate for support vector regression on the testing data was 13%. The lowest error rate of the random forest regression model made it easy for us to select it for objective trust prediction of cloud services.

4.2.2. Subjective Trust Parameters

For subjective trust computation, we have used the "Cloud Armor Trust Feedback" dataset [21]. This dataset contains 10,000+ feedbacks given by nearly 7000 consumers to 113 real-world cloud services. The feedback is based on multi-dimensional QoE parameters. The personal details of consumers are anonymized. As this dataset contains feedback for 113 cloud services, we have simulated the same number of services. The simulation parameters are listed in Table 4.

4.3. Experiment 1—Legitimate Environment

Figure 2 illustrates the trust of six cloud services in a legitimate environment when all entities are honest. The subjective trust, objective trust, and final trust are plotted in Figure 2. The notation CS_i denotes the i-th cloud service. The subjective trust is computed using Equation (4). It can be observed that the $\mathcal{T}_{sub}$ for the first cloud service, CS_1, is 0.37, CS_2 is 0.40, CS_3 is 0.26, CS_4 is 0.41, CS_5 is 0.34, and CS_6 is 0.60. The objective trust is computed using Equation (2). Likewise, the objective trust $\mathcal{T}_{obj}$ of six cloud services CS_1 to CS_6 is 0.68, 0.71, 0.62, 0.60, 0.75, and 0.66 respectively. The final trust $\mathcal{T}$ of cloud services is computed based on Equation (9). In the legitimate environment, both objective and subjective trust are assigned the same weight, i.e., $\alpha = 0.5$ in (9). A a result, the final trust of CS_1 is 0.52, CS_2 is 0.56, CS_3 is 0.43, CS_4 is 0.50, CS_5 is 0.55, and CS_6 is 0.63. To differentiate the trustworthy and untrustworthy services, an appropriate threshold is selected based on all available sample training data (that represents the ground truth) and other available information to minimise the false classifications. In this experiment, a threshold value of 0.5 was selected based on the available information and used to demonstrate how trustworthy services and untrustworthy services are differentiated. A cloud service with a final trust $\mathcal{T} \geq 0.5$ is considered trustworthy in this case. Based on the selected threshold, all services except CS_3 are considered trustworthy. A change in threshold can change such outcomes, and the threshold must be selected based on all available information to minimise the risk of false classification.

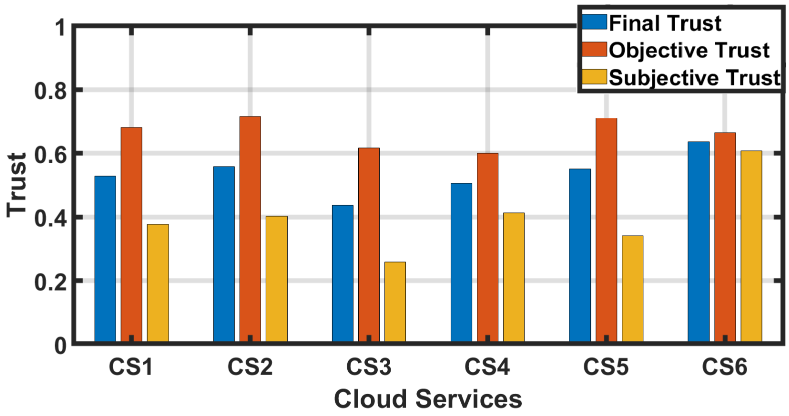


Figure 2. Cloud Services Trust.

4.4. Experiment 2—Malicious Environment

In this experiment, we will analyse the robustness of the credibility assessment module in countering the malicious behaviour of compromised CSUs. Sybil attacks are prevented by registering CSUs with unique credentials and only considering their feedback in subjective trust computation. However, for collusion attacks, five attacking scenarios are considered, whereby a percentage of the feedback given to a cloud service is assumed to be coming from malicious adversaries or compromised users. In the following, the notation S_i is used to denote the ith attacking scenario. Scenario S_1 is designed by assuming all CSUs are honest, meaning that there is no false/fabricated feedback. In the subsequent scenarios, the percentage of malicious feedback is slightly increased. For instance, in S_2, 25% of the feedback is considered to be coming from malicious CSUs. Similarly, in S_3, 50% of the feedback is assumed to be fabricated. The fourth scenario, S_4, contains 75% malicious feedback. Lastly, in S_5, all feedback is assumed to be given by malicious CSUs. In all of the collusion scenarios, the CSUs can send feedback with high values, i.e., in the range of [0.8–1], or they can send feedback with low values, i.e., between 0.05 and 0.2.

We have designed the attacking scenarios in a manner such that the credibility of user feedback can be evaluated in different configurations of hostile environments. Following this, $\mathcal{T}$ is computed in two cases. In the first case, the trust computation is based on the credibility model presented in Section 3.7. In the second case, trust is computed without a credibility evaluation. In other words, the aim of computing trust in the above-mentioned two cases is to demonstrate that our TCF can accurately compute trust by discounting the impact of malicious feedback using the proposed credibility assessment module.

Before discussing the results, it is noted that the tolerance threshold τ is assigned a value of 0.05. This is the minimum value of subjective trust variance λ that can be tolerated. Any value greater than $\tau = 0.05$ in consecutive time instances could significantly affect the trust results. It is, therefore, essential to fix a minimum value to the tolerance threshold. The trust model, by default, considers the influence of data outliers to ensure the most information-rich model is constructed with the controlled loss of some outlier data. In reality, some trustworthy feedback may be eliminated alongside the other untrustworthy feedback (as outliers) in credit assessment. This is the case of measured tolerance. As the bulk of information that separates trustworthy feedback from untrustworthy feedback is available in the clustered core regions in the data space, the contribution of a few outliers may be negligent or can be tolerated in trust modelling. If the inclusion or negligence of the outlier data points will impact the trust model, they should not be discarded.

Figure 3a illustrates the trust results of cloud service CS_5 computed without considering the credibility, i.e., the first case of collusion. In this case, CSUs are sending high feedback in all attacking scenarios. CS_5 has 994 feedbacks. Therefore, in S_1, the original feedback as found in the dataset is considered. In S_2, there are 249 high feedback values, while the rest of them are original. Likewise, in S_3, S_4, and S_5, there are 497, 746, and 949 high feedback values, respectively.

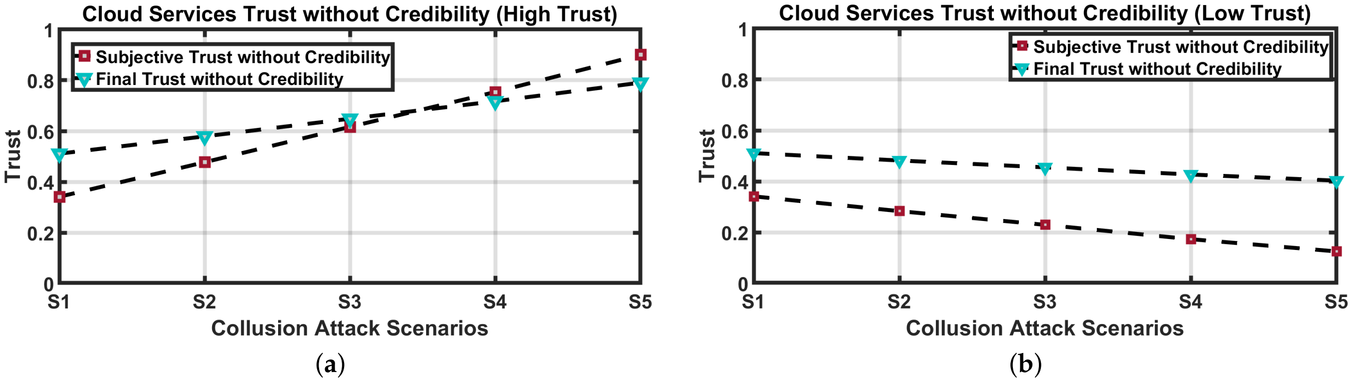


Figure 3. Cloud Services Trust without Credibility: (**a**) High Trust and (**b**) Low Trust.

It is noted that the objective trust $\mathcal{T}_{obj}$ is not changing. Therefore, the results of subjective $\mathcal{T}_{sub}$ and final trust $\mathcal{T}$ are presented. It can be observed that $\mathcal{T}_{sub}$ is increasing with an increasing percentage of high feedback. It has increased from 0.34 in S_1 to 0.9 in S_5. As a result, the final trust $\mathcal{T}$ has also increased in each attacking scenario. Comparing Figures 2 and 3a, it can be analysed that $\mathcal{T}$ is increasing in each attacking scenario. Overall, it has increased from 0.51 in S_1 to 0.79 in S_5.

In a similar fashion, we have computed the trust of CS_5 with low feedback in all attacking scenarios. Figure 3b illustrates the subjective and final trust values. It can be observed that $\mathcal{T}_{sub}$ is decreasing in each attacking scenario. In S_1, it is 0.34, in S_2, it is 0.28, and in S_3, it has further decreased to 0.22. In S_4, $\mathcal{T}_{sub}$ is 0.17, and, lastly, in S_5, it reached the lowest value of 0.12. The change in subjective trust is also affecting the final trust, which has decreased from 0.51 in S_1 to 0.40 in S_5. These experiments underline that the final trust in cloud services can be significantly increased/decreased if the credibility assessment is not carried out. Collusion attacks can dramatically change the trust of cloud services; for example, consider CS_5, which has become highly trustworthy or untrustworthy with high and low feedback.

In the second case of collusion attacks, trust is computed based on the credibility model presented in Section 3.7. As in each attacking scenario, a significant percentage of the feedback is malicious. Therefore, the credibility assessment is based on the feedback volume collusion Y parameter. It quantifies the change in subjective trust introduced by a number of collaborative CSUs. Next, $\mathcal{T}_{sub}$ in a recent time instance is adjusted based on the credibility value calculated using Equation (8). The final trust $\mathcal{T}$ of cloud services is computed using Equation (9). Nonetheless, in the case of a credibility assessment, α is assigned a different value in each attacking scenario. α is quantified based on the percentage of attacking scenarios multiplied by 0.5. Table 5 lists the α values in five attacking scenarios.

Table 5. α in different attacking scenarios.

S_1	S_2	S_3	S_4	S_5
0.5	0.375	0.25	0.17	0

Figure 4a shows the trust results of CS_5 in the case of high feedback. It can be observed that the credibility assessment model successfully prevented the malicious CSUs from dramatically increasing the trust results. Comparing the trust with the first case, Figure 3a, it can be observed that in S_2, $\mathcal{T}_{sub}$ computed with the credibility assessment is 0.32, and without it, it is 0.47. Similarly, in S_3, $\mathcal{T}_{sub}$, with credibility it is 0.43, and without it, it is 0.61. Similar patterns can be observed in the other attacking scenarios, S_4 and S_5. Moreover, due to the credibility assessment, the final trust $\mathcal{T}$ of CS_5 is also not changing dramatically. It remained between 0.51 and 0.68 in the five attacking scenarios.

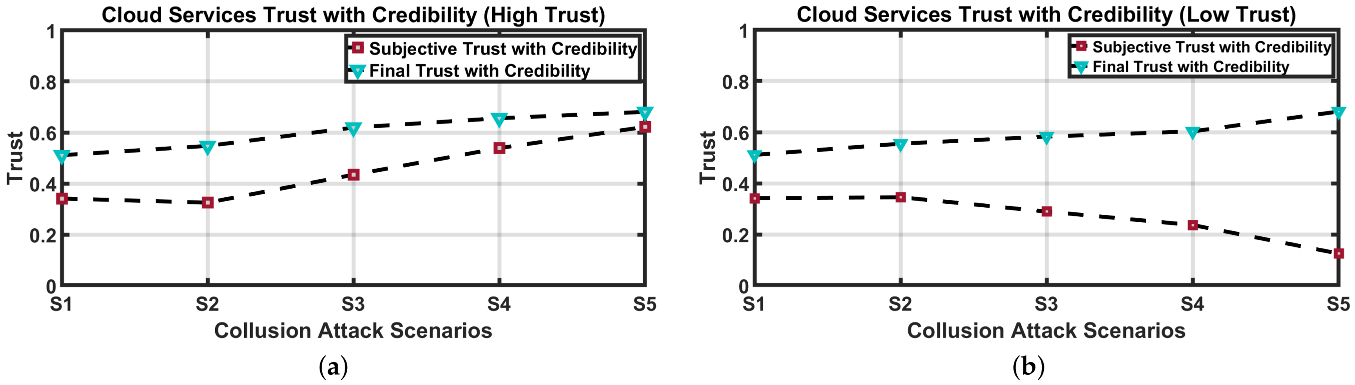


Figure 4. Cloud Services Trust with Credibility: (**a**) High Trust and (**b**) Low trust.

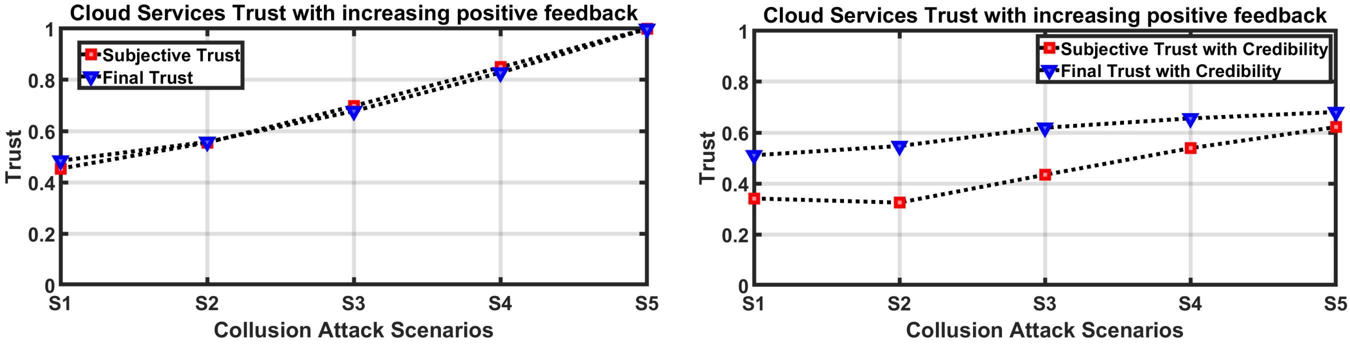
4.5. Experiment 3—Comparative Analysis

In the third experiment, we compare the results of our proposed TCF with [10], in which the subjective trust in a cloud service increases with the increasing number of positive feedback ζ and activity degree ρ. The higher number of positive feedback increases the weight of δ (i.e., Equation (16) in [10]), which eventually increases the overall trust degree (OTD). For comparison and completeness, we first computed the subjective and final trust based on the models presented in [10] and later compared them with the results obtained from our proposed TCF.

Figure 5a shows the results of CS_5 computed based on [10]. CS_5 has an activity degree $\rho = 0.98$, as it has taken up 805 jobs. Feedback greater than >0.4 is considered to be positive, and $\leq$0.4 is negative. For S_1, the number of positive feedback $\zeta = 451$, for S_2, it is 552.

Likewise, in the cases of S_3, S_4, and S_5, ζ is 694, 845, and 994, respectively. We can observe that the final trust of the cloud service CS_5 increases from 0.45 to 0.99 with the increasing number of positive feedback.

However, it is very likely that this cloud service has been affected by collusion and/or Sybil attacks. It is, therefore, essential that a trust computation model must be robust in not only detecting these attacks but also mitigating their impact. This is where the feedback credibility assessment plays its role, and we claim that our proposed approach protects against both collusion and Sybil attacks. To show the effectiveness of the credibility model, we have computed the final trust of cloud service CS_5 by taking into consideration all attacking scenarios, i.e., S_1, S_2, S_3, S_4, and S_5. Figure 5b shows the subjective and final trust of CS_5 computed based on the credibility model. The difference between the two graphs, i.e., Figure 5a,b, is quite promising. The subjective trust $\mathcal{T}_{sub}$ does not change dramatically with high feedback and remained between 0.34 and 0.62 in various scenarios. $\mathcal{T}$ also did not change much and remained between 0.51 and 0.68.



(**a**) (**b**)

Figure 5. Comparative Analysis: (**a**) Final Trust and (**b**) Final Trust with Credibility.

5. Conclusions and Future Work

Feedback is a crucial input in evaluating the trustworthiness of cloud services. Nonetheless, it is essential to evaluate the credibility of feedback before employing it to compute subjective trust. Considering the limitations of existing studies, we have proposed a new TCF in which the trust of a cloud service is computed by multi-dimensional objective and subjective evidence. For accurate trust computation and to prevent collusion and Sybil attacks, the credibility of user feedback is also evaluated.

We modelled collusion attacks in five different attacking scenarios in order to evaluate the effectiveness of the credibility assessment module. In each attacking scenario, a number of CSUs are collaborating to increase/decrease the trust in cloud services. Collaborative malicious CSUs can effectively change the trust results and achieve their aim of impacting the accuracy of TCF. Our credibility assessment based on subjective variance λ and Y precisely quantified the change in trust and adjusted it afterwards. Our proposed TCF is evaluated in legitimate and malicious environments. Our results show that the proposed TCF can successfully prevent the malicious behaviour of CSUs. Additionally, we compared our proposed trust model with [10] and showed that trust increases with positive feedback. The comparative analysis underlines that it is essential to assess the credibility of feedback based on the parameters discussed in Section 3.7.

In the future, we will further work on multi-dimensional aspects of user feedback by identifying the parameters that are considered more important by CSUs. We will investigate how a CSP can improve the QoS of cloud services and gain a high level of trust by incorporating the user-centric parameters. We believe that the prevailing scepticism regarding cloud services can also be mitigated by giving due consideration to user feedback and their preferences.

Author Contributions: Conceptualization, A.K.J. and I.A.J.; methodology, A.K.J., I.A.J. and T.J.; software, A.K.J. and I.A.J.; validation, A.K.J., I.A.J. and T.J.; formal analysis, A.K.J., I.A.J. and T.J.; investigation, A.K.J., I.A.J. and T.J.; data curation, A.K.J.; writing—original draft preparation, A.K.J. and I.A.J.; writing—eview and editing, T.J.; visualization, A.K.J.; supervision, I.A.J. and T.J.; project administration, A.K.J. and I.A.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: https://cs.adelaide.edu.au/~cloudarmor/ds.html (accessed on 24 April 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. European Parliament; Council of the European Union. General data protection regulation. In *April 2016, Regulation (EU) 2016 of the European Parliament and of the Council of on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC*; Council of the European Union: Brussels, Belgium, 2016.
2. Carey, P. *Data Protection: A Practical Guide to UK and EU Law*; Oxford University Press, Inc.: Oxford, UK, 2018.
3. Elmagzoub, M.A.; Syed, D.; Shaikh, A.; Islam, N.; Alghamdi, A.; Rizwan, S. A Survey of Swarm Intelligence Based Load Balancing Techniques in Cloud Computing Environment. *Electronics* **2021**, *10*, 2718. [CrossRef]
4. Skondras, E.; Michalas, A.; Vergados, D.J.; Michailidis, E.T.; Miridakis, N.I.; Vergados, D.D. Network Slicing on 5G Vehicular Cloud Computing Systems. *Electronics* **2021**, *10*, 1474. [CrossRef]
5. Josang, A.; Ismail, R. A survey of trust and reputation systems for online service provision. *Decis. Support Syst.* **2007**, *42*, 618–644. [CrossRef]
6. Nagarajan, A.; Varadharajan, V. Dynamic Trust Enhanced Security Model for Trusted Platform Based Services. *Future Gener. Comput. Syst.* **2011**, *27*, 564–573. [CrossRef]
7. Ferrer, A.J.; HernáNdez, F.; Tordsson, J.; Elmroth, E.; Ali-Eldin, A.; Zsigri, C.; Sirvent, R.; Guitart, J.; Badia, R.M.; Djemame, K.; et al. OPTIMIS: A Holistic Approach to Cloud Service Provisioning. *Future Gener. Comput. Syst.* **2012**, *28*, 66–77. [CrossRef]
8. Li, B.; Liao, L.; Leung, H.; Song, R. PHAT: A Preference and Honesty Aware Trust Model for Web Services. *IEEE Trans. Netw. Serv. Manag.* **2014**, *11*, 363–375. [CrossRef]
9. Li, X.; Ma, H.; Zhou, F.; Gui, X. Service Operator-Aware Trust Scheme for Resource Matchmaking across Multiple Clouds. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 1419–1429. [CrossRef]
10. Li, X.; Ma, H.; Zhou, F.; Yao, W. T-Broker: A Trust-Aware Service Brokering Scheme for Multiple Cloud Collaborative Services. *IEEE Trans. Inf. Forensics Secur.* **2015**, *10*, 1402–1415. [CrossRef]
11. Wang, Y.; Lu, Y.C.; Chen, I.R.; Cho, J.H.; Swami, A.; Lu, C.T. LogitTrust: A Logit Regression-based Trust Model for Mobile Ad Hoc Networks. In Proceedings of the 6th ASE International Conference on Privacy, Security, Risk and Trust, Cambridge, MA, USA, 13–16 December 2014.
12. Wang, T.; Li, Y.; Chen, Y.; Tian, H.; Cai, Y.; Jia, W.; Wang, B. Fog-Based Evaluation Approach for Trustworthy Communication in Sensor-Cloud System. *IEEE Commun. Lett.* **2017**, *21*, 2532–2535. [CrossRef]
13. Junejo, A.K.; Komninos, N.; Sathiyanarayanan, M.; Chowdhry, B.S. Trustee: A Trust Management System for Fog-Enabled Cyber Physical Systems. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 2030–2041. [CrossRef]
14. Fan, W.; Perros, H. A Novel Trust Management Framework for Multi-cloud Environments Based on Trust Service Providers. *Knowl.-Based Syst.* **2014**, *70*, 392–406. [CrossRef]
15. Ghosh, N.; Ghosh, S.K.; Das, S.K. SelCSP: A Framework to Facilitate Selection of Cloud Service Providers. *IEEE Trans. Cloud Comput.* **2015**, *3*, 66–79. [CrossRef]
16. Mrabet, M.; Saied, Y.B.; Saidane, L.A. CAN-TM: Chain Augmented Naïve Bayes-Based Trust Model for Reliable Cloud Service Selection. *ACM Trans. Internet Technol.* **2019**, *19*, 1–20. [CrossRef]
17. Balcao-Filho, A.; Ruiz, N.; Rosa, F.; Bonacin, R.; Jino, M. Applying a Consumer-centric Framework for Trust Assessment of Cloud Computing Service Providers. *IEEE Trans. Serv. Comput.* **2021**. [CrossRef]
18. Ahmed, U.; Al-Saidi, A.; Petri, I.; Rana, O.F. QoS-aware trust establishment for cloud federation. *Concurr. Comput. Pract. Exp.* **2022**, *34*, e6598. [CrossRef]
19. Aghaee Ghazvini, G.; Mohsenzadeh, M.; Nasiri, R.; Rahmani, A.M. A new multi-level trust management framework (MLTM) for solving the invalidity and sparse problems of user feedback ratings in cloud environments. *J. Supercomput.* **2020**, *77*, 2326–2354. [CrossRef]
20. Hassan, H.; El-Desouky, A.I.; Ibrahim, A.; El-Kenawy, E.S.M.; Arnous, R. Enhanced QoS-Based Model for Trust Assessment in Cloud Computing Environment. *IEEE Access* **2020**, *8*, 43752–43763. [CrossRef]
21. Noor, T.H.; Sheng, Q.Z.; Yao, L.; Dustdar, S.; Ngu, A.H. CloudArmor: Supporting Reputation-Based Trust Management for Cloud Services. *IEEE Trans. Distrib. Syst.* **2016**, *27*, 367–380. [CrossRef]

22. Sujatha, M.; Geetha, K.; Balakrishnan, P. User-centric framework to facilitate trust worthy cloud service provider selection based on fuzzy inference system. *J. Intell. Fuzzy Syst.* **2021**, *41*, 5629–5637. [CrossRef]
23. Chen, I.R.; Guo, J.; Wang, D.C.; Tsai, J.J.P.; Al-Hamadi, H.; You, I. Trust-Based Service Management for Mobile Cloud IoT Systems. *IEEE Trans. Netw. Serv. Manag.* **2019**, *16*, 246–263. [CrossRef]
24. Lu, L.; Yuan, Y. A novel TOPSIS evaluation scheme for cloud service trustworthiness combining objective and subjective aspects. *J. Syst. Softw.* **2018**, *143*, 71–86. [CrossRef]
25. Somu, L.; Jen, Y. A trust centric optimal service ranking approach for cloud service selection. *Future Gener. Comput. Syst.* **2018**, *86*, 234–252. [CrossRef]
26. Soleymani, S.A.; Abdullah, A.H.; Zareei, M.; Anisi, M.H.; Vargas-Rosales, C.; Khurram Khan, M.; Goudarzi, S. A Secure Trust Model Based on Fuzzy Logic in Vehicular Ad Hoc Networks With Fog Computing. *IEEE Access* **2017**, *5*, 15619–15629. [CrossRef]
27. Wang, T.; Zhang, G.; Bhuiyan, M.Z.A.; Liu, A.; Jia, W.; Xie, M. A novel trust mechanism based on Fog Computing in Sensor–Cloud System. *Future Gener. Comput. Syst.* **2020**, *109*, 573–582. [CrossRef]
28. Kliazovich, D.; Bouvry, P.; Khan, S.U. GreenCloud: A packet-level simulator of energy-aware cloud computing data centers. *J. Supercomput.* **2012**, *62*, 1263–1283. [CrossRef]