

Design of an Efficient Trustful-Lightweight Cloud Service Provisioning Model using Service Optimizer

Suvarna S. Pawar^{1,*}, Y. Prasanth²

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation
Vaddeswaram - 522502, Guntur District, Andhra Pradesh, India

Abstract

INTRODUCTION: The establishment of trusted cloud services pretends to provide high impactful service with better satisfaction to the web-users and cloud service providers. Moreover, various existing trust-aware algorithms use diverse QoS measurements and related attributes, leading to the complex selection of cloud services.

OBJECTIVES: Thus, this research intends to propose a Trustful-Lightweight Cloud Service Provisioning algorithm (TL-CSP) using Service Optimizer (SO).

METHODS: Initially, the QoS metrics are determined by evaluating attributes based on a ranking method based on the users' requests. It is performed with the computation of weighted coefficients of received requests from the users. The service optimization is performed using a global optimizer to assist the cloud users in selecting the service with better satisfaction.

RESULTS: The proposed TL-CSP accuracy is validated and compared with the existing cloud service provisioning algorithm to measure the proposed model's efficiency.

CONCLUSION: The simulation is carried out in a MATLAB environment. The proposed TL-CSP intends to show a better trade-off in contrast to prevailing approaches.

Keywords: Cloud services, QoS, trust-aware model, Service Optimizer, Trustful Lightweight Cloud Service provisioning, weighted coefficients

Received on 06 June 2021, accepted on 18 October 2021, published on 19 October 2021

Copyright © 2021 Suvarna S. Pawar *et al.*, licensed to EAI. This is an open access article distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/3.0/>), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/eai.19-10-2021.171468

1. Introduction

Optimization is defined as “The services offered by the cloud acts as application-level modelling in various real-time environments. The IT resource utilization and delivery of essential services to attain better performance in software applications, infrastructure, and platforms via the cloud-deployed environment [1]. This environment is set during the need for scalability and on-demand requests [2]. The cloud computing (CC) environment is deployed based on hybrid applications, end-users, and internet provisioning [3]. It enhances the storage capability and computational efficiency over the terminal regions and offers the end-users with resourceful and better functional experiences [4].

But, there exists both merits and de-merits in CC based on trust establishment, service provisioning, resource utilization, security, and so on [5]. The preliminary stage of cloud functionality is initiated with the trusted establishment as the information is gathered from various resources [6]. The essential constraints related to available resources and trust establishment are uncertainty and openness, which causes higher un-stability and uncertainty measures to fulfil security and Quality of Services (QoS) [7]. However, with the complex service provisioning, the process of attaining effectual service composition and integrating the services with credibility has turned to be the essential issues over the field of CC research [8]. There are diverse valuable service composition and task scheduling approaches anticipated for the conventional

*Corresponding author. Email: sspawar.scoe@sinhgad.edu

network environment [9]. Moreover, the prevailing internet provisioning method lags in establishing active collaboration with the users involved in the computing process [10]. For this cause, some investigators concentrate on modelling an agent-based CC model [11] – [12]. The cloud system with a multi-agent-based architectural model is hugely easier for projecting cloud entities' initiation, intelligence, and autonomy for realizing the autonomous evolution of various cloud services, which is much nearer to the commercial market nature [13]-[14].

This research concentrates on modelling and efficient Trustful-Lightweight Cloud Service Provisioning algorithm (TL-CSP) with service optimizer (SO) to fulfil the CC environment's requirements mentioned above. Here, a novel trustful service provisioning model is provided with the optimization of services. The proposed TL-CSP facilitates trust-enable service provisioning to accelerate customer service preferences for categorizing the service preferences. The service optimizer is equipped with a specific two-stage process to enhance the success rate and QoS fulfilment [15]. To project the significance of the proposed TL-CSP as a reliable, satisfactory, and efficient model, the following queries need to be resolved appropriately. They are:

- 1) Which framework is more appropriate for trust-establishment among the connected nodes? How these nodes interact with one another?
- 2) Which model is more suitable for trust-management as trust is concerned with the context-aware model?
- 3) What algorithm is best suited for learning the user's need and service preferences?

In contradiction to prevailing approaches, this research concentrates on the impact of trustful service provisioning model establishment over the cloud environment. The significance of this research is listed below:

- 1) To model an efficient Trustful-Lightweight Cloud Service Provisioning algorithm (TL-CSP) model with service optimizer.
- 2) To fulfil the QoS requirements by measuring the attributes related to the trust-establishment model.
- 3) To perform computation using weighted coefficients of received requests from the users with the knowledge learned from the user service request and priorities. Here, some experimental analysis is done to test and validate the significance of the proposed -Lightweight Cloud Service provisioning algorithm (TL-CSP) model over the cloud environment.

The work is further partitioned as follows: section 2 gives a brief description of the trust establishment model's background knowledge. Section 3 provides the system model with elaborating the proposed -Lightweight Cloud

Service Provisioning algorithm (TL-CSP) model with service optimizer. Section 4 is set with the experimentation process with numerical results and discussions. Section 5 is a conclusion that shows the limitations of the proposed model with the idea of future research enhancements.

2. Related work

The idea behind the proposed Lightweight Cloud Service Provisioning algorithm (TL-CSP) model is triggered by the various background knowledge extracted from the extensive review. Here, the analysis is done with multiple trust establishment and management models in CC. Diverse cloud-service brokers/systems are emerging to handle various issues like privacy, security, trust, etc. Rauchwerger et al. [16] design a lattice monitoring approach for cloud service management. This monitoring process is the preliminary requirements of internet elements, specifically over cloud services. It includes both service and infrastructure management. The author concentrates on various issues that are related to cloud service management. The significant features of reservoir projects are discussed with proper design methods and implementation. There are diverse outcomes that are generated to meet out the issues related to a cloud environment. However, there is a considerable gap when it comes to cloud monitoring and management. To deal with this issue, the author concentrates on experimentation with the private cloud using case studies and application design. The significant findings show that there is a possibility to deploy the private cloud environment with the necessary tools. De Chaves et al. [17] provide a web-based model for dealing with the cloud infrastructure from various service providers. It facilitates easier deployment and handling of critical-business applications with private, public, and hybrid cloud.

Similarly, Spring et al. [18] discuss the cloud capacity by providing various services with added limits and providing benefits to the providers with fewer computation costs. Li et al. [19] discuss the brokering concepts in an independent manner devoid of validating service providers' cost. Therefore, optimization approaches are used to place VM over the appropriate rate of service providers.

Various research groups work on a trust management environment in a cloud computing environment. Qi et al.[20] perform an extensive review with trust requirements over the cloud system regarding privacy and security. The service providers need to take the necessary actions to improve the trust over the services and the providers. It

includes protection, prevention, identified control, ownership, and critical concepts that determine the user's trust level based on the provided service. The transparency lack and control reduction are the major issues that have to be concentrated on. The author needs to identify the remote access control for users, transparency towards cloud users over automatic traceability. The certification for security is used to improve the trust model. Wahab et al. [21] use the trusted-overlay model to execute system reputation for determining trust among the data owners and service providers. The distributed hash-table among the virtualized environment is constructed with the reputation system. The networks are provisioned for trust management and security enforcement model. Some methods like watermarking and coloring are used to preserve the software modules and data objects. The multi-way authentication facilitates the cloud users to evaluate the sensitive data over the private and public cloud. Sasiet al. [22] provide an integrated architectural model for security reinforcement and privacy over the cloud environment. Here, the virtual cluster integration is performed where the reputation system performs the trusted data access. Moreover, the author concentrates on privacy and authorized issues over the end-user side but lacks in server-side protection model.

Sherubha et al. [23] initiate a novel trust model to measure the resources using resource brokers. The broker needs to select appropriate cloud resources over a heterogeneous environment based on the user's requirements. The anticipated trust management model is executed with an infra-structure model and authentication approach. The trust model computes the trust value among the resources and the behavioral trust model. Moreover, two simple models are used to quantify the resource trustworthiness, which causes unfair and incomplete trust-based decision-making process. This dynamic model shows business and social relationships among the trusted mechanism, including active, real-time, security, and reliability. Li et al. [24] anticipate a novel trust-based model for a multi-tenant environment. It helps in resolving the issues over trust management with the sum of trustful-service providers. This model is based on trust propagation and trust evaluation model.

Moreover, the trust evaluation model utilizes the conventional probability method indeed of direct service behavior. It lags in the direct and indirect fusion of the trust model. Liu et al. [25] anticipate a selective cloud service provider to offer better service provisioning. It merges the competency and trustworthiness to compute the interaction risk. It is evaluated from various experiences attained from direct interactions to feedback reputation. However,

competency is measured with the transparency towards the SLA guarantees. Also, the trusted evaluation makes use of conventional subjective rating with real-time service characteristics. Author et al., [] anticipates a trust-aware model validates the security control over user's requirements. It initiates the property taxonomy relies on semantic and predicting authorities for property validation. The taxonomies rely on trust formulation to ensure consumers enable cloud providers' trustworthiness. It is based on a rating model.

Movahedi et al. [26] anticipate a service mechanism with trust improved secure model for evaluating trusted platforms. It deals with the nature of both property and binary-based attestation method, which cannot be attesting absolutely for the service characteristics. It utilizes a hybrid-trust model to integrate trust from properties, measurements, and previous recommendations and experiences to diminish uncertainties. Moreover, it fuses with trust-factors, for instance, soft Trust or hard Trust. Wang et al., [27] trust as a Service structure is used to enhance trust management. Specifically, the author initiates an adaptive credibility model that differentiates negative and credible feedback by determining cloud service customers and the feedback. Also, it does not facilitate trustworthiness based on user's feedback and monitoring information. Basciftci et al. [28] concentrate on distributed agents, cloud service framework is recommended for effectual scheduling and diverse user request. The architecture attempts to examine servers dynamically and facilitates high-quality evaluating users' resources. The mechanisms make use of monitored information with the large cloud environment. The data comprises certain specific details, response time, and resource utilization [29] and examines the appropriate resources and information on every occasion, and facilitates immediate user requests. As an outcome, the system offers higher trustworthiness and services by reviewing the data and probably using it efficiently [30]. The author focuses on service operator-based trust establishment during the resource match-making over the cloud environment. Thus, the outcomes provide a direct information monitoring system with essential feedback data.

3. Methodology

The trust model is depicted as establishing trust among the trustee and trustor for recognition of trustee identity to perform the specific particular task over some time. It is a kind of decision enacted by the trustor which relies

on the experience and other knowledge. The trust indicator is composed of stability, honesty, reliability, and authenticity. The traditional trust model is generally composed of recommender, trustor, and trustee. The trustor measure the trust among the transactional entities (users/nodes), the trustee is adopted for the transaction and selection process. The recommender establishes the relationship between the trustor and trustee, as in Fig 1.

The agent over the cloud environment is used for selecting the communication and cooperation among various users. The agent needs to be placed in a proper environment and sense the environmental variation. It comprises essential components with an intelligent agent

that includes a communication module, message processing unit, and sensor modules. These modules help preserve the environmental variations by handling message processing and communication to assist the agents in learning the preference transactions. The agents have to generalize the characteristics based on feedback from the cloud environment. This module is no longer confined to pre-defined behaviours. The proposed Trustful- Lightweight Cloud Service Provisioning algorithm (TL- CSP) model is equipped with the service optimizer to manage the trustful service provisioning mechanism. Thus, this model enhances the stability and orderliness during information transmission.

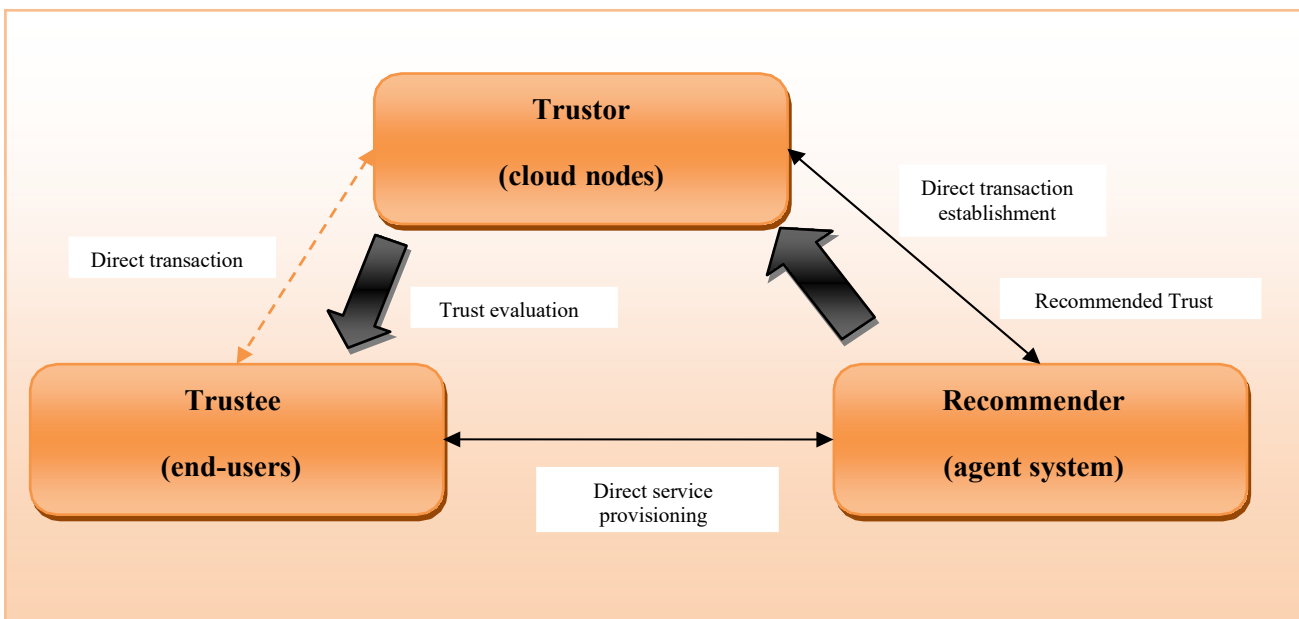


Fig 1. Traditional trust model

When direct transactions among the users are performed, there is a trustful establishment among the cloud agents and the end-users. Else, the recommender is involved in establishing the relationship. The trusted identity is established during network initialization. Hence, the evaluation among these agent models is expressed as in Eq. (1):

$$T_{sp}^{(A \rightarrow B)} = \alpha T_d^{(A \rightarrow B, sp)} + \beta T_r^{(A \rightarrow B, sp)} \quad (1)$$

Here, $T_{sp}^{(A \rightarrow B)}$ is the trustful entity among the cloud with the trust established trading context. It helps direct trust establishment among A and B, α , and β specifies the weight of immediate and recommendation trust. It is a context-sensitive model to fulfill the QoS of a cloud entity, different from other services. The accuracy of the proposed Trustful-Lightweight Cloud Service Provisioning model relies on the cloud-based trust establishment. Cloud service provisioning is performed with three diverse service types known as storage, computation,

and network. $CSP_{trustfulness} = \{T_{cpu}, T_{storage}, \text{ and } T_{bandwidth}\}$ These three factors determine the trustee of analysis, storage, and network, respectively.

3.1. Trust evaluation with service optimizer

The term service optimizer acts as a decision-making process for handling multiple factors. The service optimizer holds certain features towards subjectivity based on service values and attributes, uncertainty (changing cloud environment), provisioning sensitivity (Trust is different for providers and services), and multiple QoS factors. It is more appropriate to make use of a service optimizer to evaluate the service provisioning tasks. However, the SO theory handles complex problems to deal with multiple QoS factors. Thus, this research adopts the SO concept to assess the trust establishment process (See Fig 2). The steps performed by the service optimizer are given below:

- 1) In SO, there are three diverse trust evaluation process, storage, network, and computation with a set of evaluation factors given as $\{EF_1, EF_2, \dots, EF_3\}$.
- 2) The determination of evaluation level $EL = \{EL_1, EL_2, \dots, EL_n\}$. EL specifies the level or values of each factor. It selects four levels for trust establishment: 'highly trustful,' 'nominal trustful,' 'nominal un-trustful,' and 'un-trustful model' to specify the trust degree of users on the service.

3) The weighted factors are determined as $W = \{w_1, w_2, \dots, w_m\}, w_i > 0, \sum w_i = 1$. Here, w_i is attained with the influence of EF_i to determine the overall decision process and various factors' weight that is customized based on service preferences.

4) A matrix format is constructed for deciding during the appropriate time for all factors to evaluate the degree. The computation of the evaluation factors is based on the matrix format. The relationship among the agents and the users are reviewed with factors as expressed below in Eq. (2):

$$M = \begin{bmatrix} m_{11} & \dots & m_{1m} \\ \vdots & \ddots & \vdots \\ m_{n1} & \dots & m_{rn} \end{bmatrix} \quad (2)$$

The frequency model is utilized to determine the 'M' value. The trustful value is categorized based on various evaluation level y partitioning the intervals, historical data frequency based on provided intervals. This process is considered too intensive and does not pretend to increase the overhead during the decision-making process. The complete weighted average is expressed as in Eq. (3):

$$\text{weighted average } W_a = W_n \cdot M \quad (3)$$

The direct trust evaluation is performed with a direct trust relationship, which is attained during direct interaction among the trust establishment partners (end-users and nodes). $T_d^{(A \rightarrow B, tc)}$ ($T_d^{A \rightarrow B, tc} = \{t_{d1}, t_{d2}, \dots, t_{dm}\}$) is utilized to specify

the direct trust model for all evaluation factor. The computation is based on Eq. (4):

$$t_{di}^{A \rightarrow B, tc} = \frac{TNum_{t \in E}^{lia \rightarrow B, sp}}{TNum_{total}^{A \rightarrow B, sp}} \quad (4)$$

From Eq. (4), $TNum_{t \in v_i}^{A \rightarrow B, sp}$ specifies the total transactions among the trust evaluation process that belongs to E_{li} among two entities A and B with service provisioning factor, (sp). It makes use of trustful recommendation. The trust recommendation process is provided as $T_m^{C \rightarrow B, sp}$ ($T_m^{C \rightarrow B, sp} = \{t_{m1}, t_{m2}, \dots, t_{mn}\}$) where the degree of evaluation and SO proves two types of recommendation. They are:

- 1) end-users selects the reliable recommender;
- 2) computes the trust recommendation process based on recommendation value and weight, respectively. It is expressed as in Eq. (5):

$$T^{NA-B,sp} = \frac{\sum P \in \Omega (T_d^{A \rightarrow C, recommendation} * T_d^{c \rightarrow B, sp})}{length(\Omega)} \quad (5)$$

Here, $T_d^{c \rightarrow B, sp}$ is direct trust recommendation in context with service provisioning, $T_d^{A \rightarrow C, the recommendation}$ is trust over the recommender. Ω specifies entities over the cloud, $length(\Omega)$ is recommendation set size respectively. The computation method of $T_d^{A \rightarrow C, the recommendation}$ is the same alike of direct trust-based transaction. Based on this, Eq. (6) is expressed as:

$$t_{di}^{A \rightarrow C, recommendation} = \frac{TNum_{t \in EV^A}^{A \rightarrow B, recommendation}}{TNum_{total}^{A \rightarrow C, recommendation}} \quad (6)$$

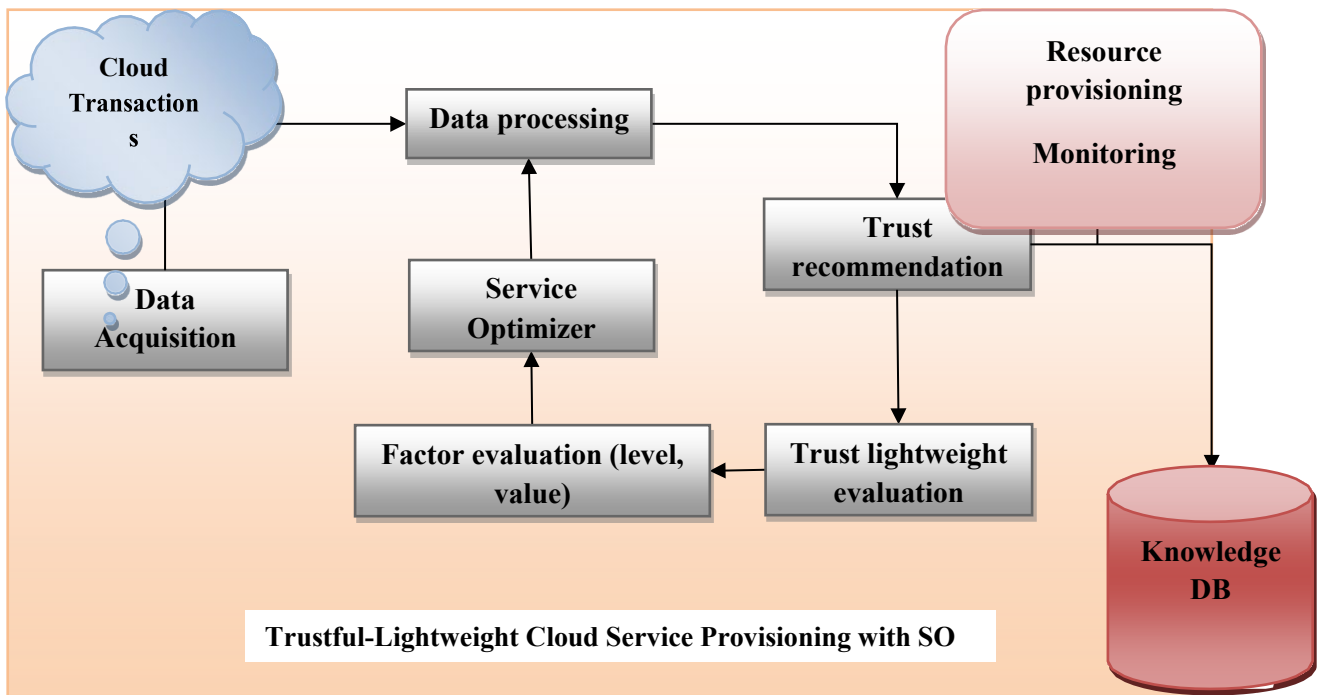


Fig 2. Trustful-lightweight cloud service provisioning-SO

Algorithm 1: Trustful Lightweight Cloud Service Provisioning- SO

1. Initialize Trust_service (agent_list, service_type, evaluation factor)
2. While service provisioning request queue =) do
3. Initiate the service request from the provided queue;
4. Compute request processing;
5. If (request = service provisioning), then
6. Predict the best fit provisioning
7. Suggest trustful service to the end-users;
8. Else
9. Request agents for further request;
10. Await for the response from agents;
11. If response is received/reaches the deadline, then
12. Predict the agents based on evaluation factors/value;
13. End if
14. If (agent success), then
15. Send a trustful recommendation then.
16. Else
17. Reject the users' request
18. End if
19. End if
20. End while
21. End initialized process

2. If trust evaluation fulfilled then
3. If trust evaluation > decision_making then
4. Trust_evaluation
5. (user, provider) → better_service;
6. Else
7. Trust_evaluation
8. (user, provider) → worst_service;
9. End if
10. End if

End trust evaluation process

6. Trust_evaluation(user, provider) → worst_service;
7. End if
8. End if
9. End trust evaluation process

There is three diverse service provisioning model known as a recommendation, trust evaluation, and service provisioning. The service recommendation assists the users' in predicting the suitable (best, worst) service. Moreover, the service is provided by the agents and the lightweight recommendation system. The service provisioning model deals with service provision and actual transactions among the users, providers, and agents. The evaluation is done with feedback from the users and constructs the training set for further evaluation.

3.2 Lightweight service provisioning (LSP)

LSP deals with user's service request, which needs agents in successive stages with service availability. The service provisioning relies on kernel functionality. In the initial stage, the agents need to transfer a request to well-known agents. These familiarized agents are known as Lightweight agents for providing services. When the agent receives the request and handles all the requests, it provides services to the end-users directly.

Moreover, when it cannot receive any request, it generates the request to agents to further process and transverse the response to the users'. When all the end-users' select the best response and transfers the 'ACK' message and 'REJECT' message to others, the agents need to ask the service providers for further confirmation and assists the end-users for further processing. After all the evaluation process, the entities need to provide feedback for superior service selection. The agent with recommendation abilities is learned by the user's service provisioning priority (rank) and varies the service provisioning based on the environmental conditions.

4. Results and Discussion

The simulation is performed in MATLAB environment over 64 bits PC with 16 GB RAM and CPU Intel core i7 processor to evaluate metrics like QoS, response time, throughput,

//Trust_evaluation

1. Initialize trust_evaluation (decision-making)



Mean Absolute Error, cosine similarity, Karl Pearson correlation coefficient, and PCC (Pearson Correlation Coefficient). The validation is done with an online available QoS dataset [31] for response time evaluation and throughput, respectively. The performance metrics are discussed below:

1) Mean Absolute Percentage Error (MAPE)

$$MAPE = \frac{1}{N} \sum_{i=1}^n \frac{|y_i - y'_i|}{y_i} \tag{7}$$

Here, 'N' is several prediction results, y'_i and y_i are predicted and real values, respectively. It shows the expected value degree, which deviates from actual values.

2) Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{N} \sum_{k=1}^N \left(\frac{y_i - y'_i}{y_i} \right)^2} \tag{8}$$

It not only reflects the magnitude of relative error; however it shows the stability towards prediction values.

3) Throughput

Throughput is the actual rate of information transferred. It is depicted as the data quantity being received or sent over unit time.

4) Pearson correlation coefficient (PCC)

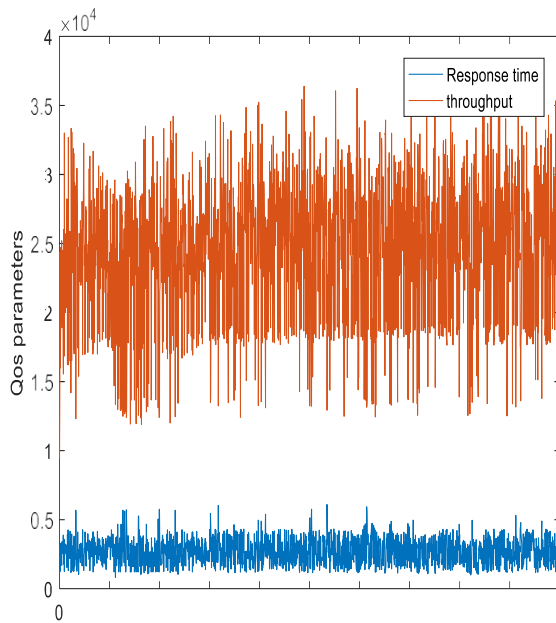


Fig 3 CS versus QoS parameters

PCC is a kind of correlation coefficient that specifies the relationship among the evaluated variables based on the ratio scale or same interval. It is evaluated with the strength of association among two continuous variables.

$$r = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2 \sum(y_i - \bar{y})^2}} \tag{9}$$

5) Karl Pearson's coefficient of correlation (KPCC)

The symbol $\rho(X, Y)$ represents it. It is related to the correlation coefficient derived by the mean of $\rho(X, Y)$. It is expressed as:

$$r = \frac{\sum d_x d_y}{\sqrt{d_x^2 \cdot \sum d_y^2}} \tag{10}$$

In the proposed model, the effect of the trust towards the transaction success rate of the service. The trust preserves the transactional success rate at a relatively higher level when there are malicious service providers. The trust decision is carried out for all transactions, which fulfils cloud users with credible partners. It helps to eliminate false transactions or invalid process. The QoS metrics are analysed in terms of response time and throughput, as shown in Fig 3. Fig 4 depicts the graphical representation of MAE where the mean absolute error is computed for 2000 users for cloud service 1. Fig 5 and Fig 6 show users' evaluation versus MAE for cloud services 2 and 3, respectively.

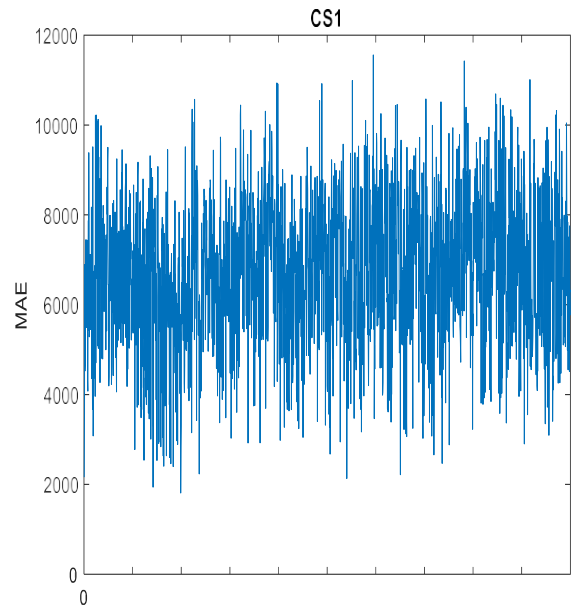


Fig 4 Users versus MAE (CS 1)

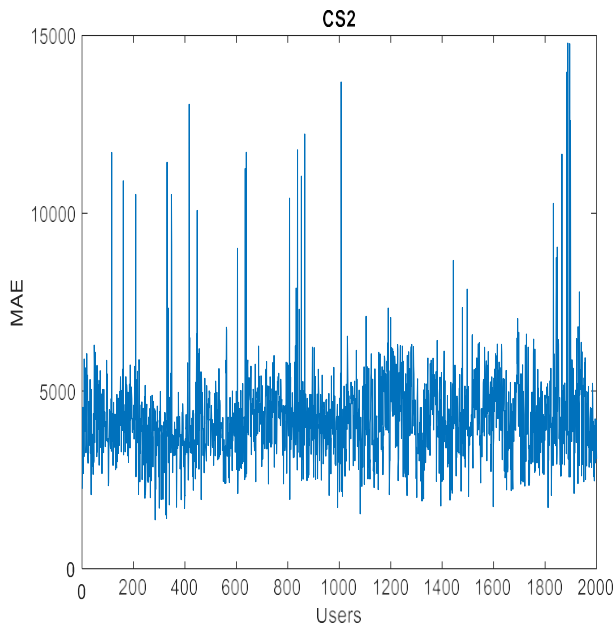


Fig 5 Users versus MAE (CS 2)

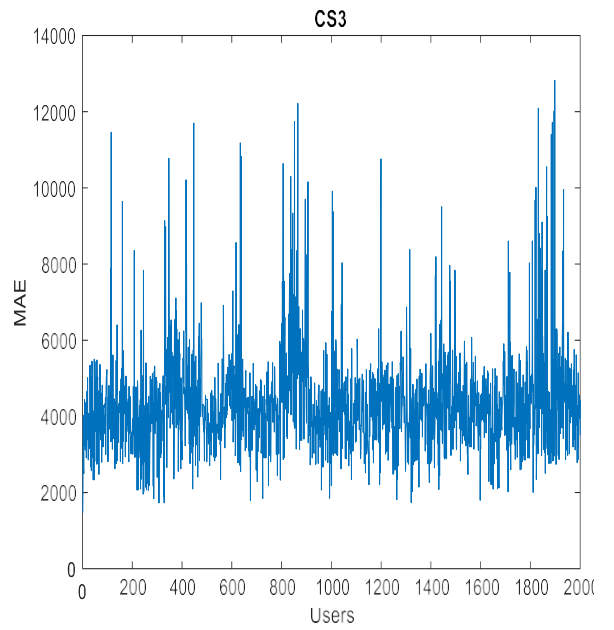


Fig 6 Users versus MAE (CS 3)

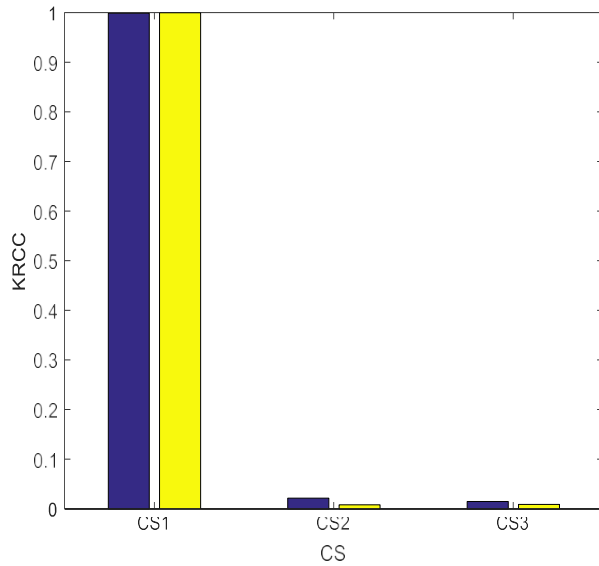


Fig 7 KRCC computation of CS 1-3

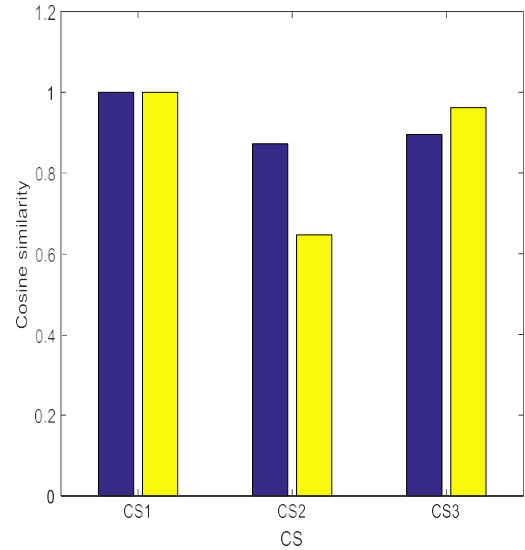


Fig 8 Cosine similarity measure of CS 1-3

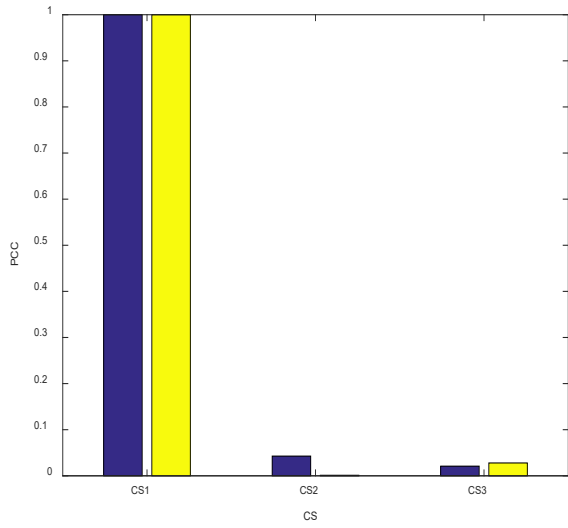


Fig 9 PCC of CS 1-3

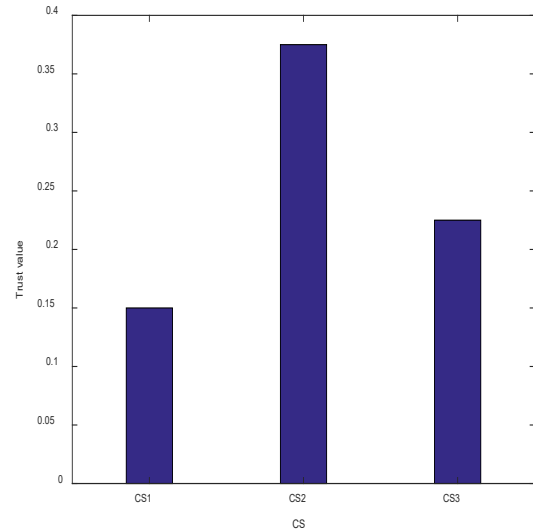


Fig 10 Trust value establishment of three different cloud services

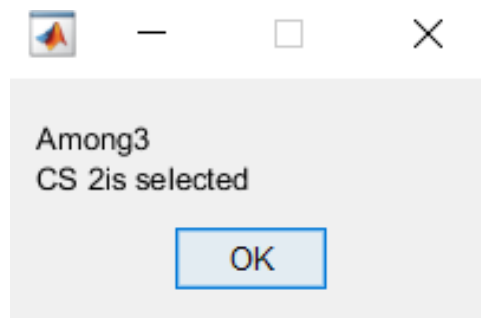


Fig 11 CS2 selection process

When more number transactions are performed, and the agents learn the transaction requirements, the accuracy is improved by preserving a higher success rate and user's fulfilment. Moreover, with the provided time, the users' service preference is pronounced with various transactions gradually. The evaluation process is satisfied based on transaction samples with positive and negative samples. Threshold levels are set to provide service satisfaction. The impact of various threshold levels over the success ratio of service provisioning and user's satisfaction is attained with a higher success rate and fulfilment. A relatively lower threshold level facilitates the agents to initiate service recommendations and resources more actively, which results in a higher success rate and satisfaction.

Moreover, when a higher threshold level is attained, more rigorous resource allocation needs to be offered for a higher success rate. Moreover, when the threshold value is higher, it reduces the number of samples for service optimization. Thus, the success rate and the satisfactory levels have to be relatively closer. The cloud service models' correlation coefficient is evaluated with the available cloud service model, where the transactions are analyzed to fulfil the QoS metrics. Initially, KRCC is considered for CS1, CS2, and CS3, where the preference of CS1 is higher when compared to CS2 and CS3 (See Fig 7). Fig 8 depicts the cosine similarity measure of CS1, CS2, and CS3, where the preference of CS1 and CS3 is higher when compared CS2. Similarly, PCC of CS1, CS2, and CS3 is examined in Fig 9, where the preference of CS1 is higher than CS2 and CS3.

Thus, the trust value is computed for all three services. Based on the observations, it is noted that CS1 and CS3 are given higher service than CS2. Thus, more resources are allocated to CS1 and CS3, which causes complexity during further transactions; therefore, the higher chances of additional trades are moved to CS2. Hence, CS2 is chosen for the different processes as depicted in Fig 11. The trustfulness of CS2 is higher to provide further cloud services in a lightweight (ranking) manner using the service optimizer.

5. Conclusion and Future Work

This research concentrates on a novel Trustful- Lightweight Cloud Service Provisioning algorithm (TL- CSP) using Service Optimizer (SO) for establishing trust model over the cloud environment. The relationship between trust and service provisioning is more critical in the cloud. Initially, the service optimizer's functionality is discussed, along with the cloud service provisioning with the lightweight model is discussed. The anticipated Trustful-Lightweight Cloud Service Provisioning algorithm (TL-CSP) using Service

Optimizer (SO) model gives a better trade-off between the trust establishment and service provisioning. The experimental outcomes show that the research provides efficient service provisioning results by quantifying service preferences, service, trust models, etc. The constraint encountered here is the lack of privacy and fault tolerance discussion while performing trust establishment. In the future, efficient fault tolerance must be concentrated to provide a unique framework to mitigate the relationship among the trust, privacy, and fault tolerance. It offers an effective mechanism to maintain confidentiality and avoids faulty conditions over the connected

References

- [1] Ma, H. Zhu, Z. Hu, W. Tang, and P. Dong, "Multi-valued collaborative QoS prediction for cloud service via time series analysis," *Future Generation Computer Systems*, vol. 68, no. 3, pp. 275-288, 2017.
- [2] Ma, H. Zhu, Z. Hu, K. Li, and W. Tang, "Time-aware trustworthiness ranking prediction for cloud services using interval neutrosophic set and ELECTRE," *Knowledge-Based Systems*, vol. 138, pp. 27-45, 2017.
- [3] Ma, Z. Hu, K. Li, and H. Zhang, "Toward trustworthy cloud service selection: A time-aware approach using interval neutrosophic set," *Journal of Parallel and Distributed Computing*, vol. 96, pp. 75-94, 2016.
- [4] Zhang, Z. Zheng, and M. R. Lyu, "WSPred: A time-aware personalized QoS prediction framework for web services," in *Proc. of the 2011 IEEE 22nd International Symposium on Software Reliability Engineering (ISSRE)*, 2011, pp. 210-219.
- [5] Syed, A. Gani, R. W. Ahmad, M. K. Khan, and A. I. A. Ahmed, "Cloud monitoring: A review, taxonomy, and open research issues," *Journal of Network Computer Applications*, vol. 98, pp. 11-26, 2017.
- [6] Zheng, X. Wu, Y. Zhang, M. R. Lyu, and J. Wang, "QoS ranking prediction for cloud services," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1213-1222, 2013.
- [7] Chintala, R.R., Narasinga Rao, M.R., Venkateswarlu, S. "Performance metrics and energy evaluation of a lightweight block cipher in human sensor networks", *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8 , no. 4 ,pp. 1487-1490, 2019.
- [8] Sambasiva Rao, K., Kameswara Rao, M. "A lightweight digital signature generation mechanism for authentication of IoT devices ", *International Journal of Recent Technology and Engineering*, vol. 7, no. 6, pp. 1862-1866, 2019.
- [9] Noora, S. Zeadallyb, A. Alfazic, and Q. Sheng,

- “Mobile cloud computing: Challenges and future research directions,” *Journal of Network and Computer Applications*, vol. 115, pp.70–85, Aug. 2018.
- [10] Lin, W.Guo, and G. Chen, “Scheduling strategy for science workflow with deadline constraint on multi-cloud,” *Journal on Communications*, vol.39, no.1, pp.56-69, Jan. 2018.
- [11] Kadham, N.R., Sreenivasa Ravi, K. “A lightweight One Time Password (OTP) based smart learning in Internet of Things”, 2018 *International Journal of Engineering and Technology(UAE)*, vol. 7 pp.480-483, 2018.
- [12] Alsaghier, H.M., Shakeel Ahamad, S., Udgata, S.K., Reddy, L.S.S. “A secure and lightweight protocol for mobile DRM based on DRM community cloud (DCC)”, *Advances in Intelligent Systems and Computing*, vol. 515, pp.475-483, 2018.
- [13] Yin, L. Chen, Y. Xu, and J. Wan, “Location-Aware Service Recommendation With Enhanced Probabilistic Matrix Factorization,” *IEEE Access*, vol. 6, pp. 62815-62825, Sept. 2018.
- [14] Yin, F. Yu, Y. Xu, L. Yu, and J. Mu, “Network Location- Aware Service Recommendation with Random Walk in Cyber- Physical Systems,” *Sensors*, vol. 17, no. 9, pp. 2059, Sept. 2017.
- [15] Abdallah, M. Zulkernine, Y. Gu, and C. Liem, “TRUST- CAP: A Trust Model for Cloud-based Applications,” in *Proc. of the 2017 IEEE 41st Annual Computer Software and Applications Conference*, Turin, Italy, 2017, pp. 584-589.
- [16] Swathi, D.R., Srikanth, V. “Hybrid trust management in cloud computing”, *International Journal of Advanced Science and Technology*, vol. 29, no. 3, pp. 8394-8401,2020
- [17] Potluri, S., Rao, K.S. “Simulation of QoS-Based Task Scheduling Policy for Dependent and Independent Tasks in a Cloud Environment”, *Smart Innovation, Systems and Technologies*, vol. 159, pp.515-525,2020
- [18] Sirisha, N., Kiran, K.V.D., “An efficient and lightweight security scheme for big data”, *International Journal on Emerging Technologies* vol. 11 , no. 1, pp. 414-420, 2020.
- [19] Li, F. Zhou, and J. Du, “LDTS: A lightweight and dependable trust system for clustered wireless sensor networks,”*IEEE Trans. Inf.Forensics Security*, vol. 8, no. 6, pp. 924–935, Jun. 2013.
- [20] Gali, S., Nidumolu, V. “Multi-context trust aware routing for internet of things”, *International Journal of Intelligent Engineering and Systems*, vol. 12, no. 1, pp. 189-200,2019.
- [21] Wuhib and R. Stadler, “Distributed monitoring and resource management for large cloud environments,” in *Proc. IFIP/IEEE Int.Symp. Integer. Netw. Manage. (IM)*, May 2011, pp. 970–975.
- [22] Sherubha, "Clone Attack Detection Using Random Forest and Multi-Objective Cuckoo Search Classification," *IEEE International Conference on Communication and Signal Processing*, April 4-6, 2019, India.
- [23] Dhote, B.L., Krishna Mohan, G. “Trust and Security to Shared Data in Cloud Computing: Open Issues, *Advances in Intelligent Systems and Computing*”, vol. 870, pp. 117-126, 2019
- [24] Patil Abhijit, J., Syam Prasad, G.,”Trust based security model for IoT and fog based applications, *International Journal of Engineering and Technology(UAE)*”, vol. 7, pp. 691- 695,2018
- [25] Nashnosh, M.T., Vijaya Babu, B.,” LESIPT: Lightweight enhanced secure incentive protocol with trusted value for multi- hop wireless networks” ,*International Journal of Applied Engineering Research* vol. 9, no. 19, pp. 5373- 5384, 2014
- [26] Kameswara Rao, M., Usha Switha, T., Naveen, S. “A novel graphical password authentication mechanism for cloud services”, *Advances in Intelligent Systems and Computing*, vol. 433, pp. 447-453,2016.
- [27] Wang, J. Zhang, and Y. Wang, “Trust-aware privacy- preserving recommender system,” in *Proc. 9th EAI Int. Conf.Mobile Multimedia Commun.*, 2016, pp. 107–115
- [28] Basciftci, Y. Wang, and P. Ishwar, "On privacy-utility trade-offs for constrained data release mechanisms," in *Proc.Inf. Theory Appl. The workshop*, Jan./Feb. 2017, pp. 1–6
- [29] Nishad, J. Paliwal, R. Pandey, S. Beniwal, and S. Kumar, “Security, privacy issues and challenges in cloud computing: A survey,” in *Proc. 2nd Int. Conf. Inf. Commun. Technol. Competitive Strategies*, 2016, p. 47.
- [30] Zhu, “Privacy calculus and its utility for personalization services in E-commerce: An analysis of consumer decision- making,” *Inf. Manage.*, vol. 54, no. 4, pp. 427–437, 2017.
- [31] Laghari, A. A., He, H., Khan, A., Kumar, N., & Kharel, R. (2018). Quality of experience framework for cloud computing (QoC). *IEEE Access*, 6, 64876-64890.
- [32] Laghari, Asif Ali, Awais Khan Jumani, and Rashid Ali Laghari. "Review and State of Art of Fog Computing." *Archives of Computational Methods in Engineering* (2021): 1-13.
- [33] Yadav, Rahul, Weizhe Zhang, Keqin Li, Chuanyi Liu, and Asif Ali Laghari. "Managing overloaded hosts for energy- efficiency in cloud data centers." *Cluster Computing* (2021): 1- 15.
- [34] Karim, Sajida, Hui He, Arif Hussain Magsi, and Rashid Ali Laghari. "Quality of service (QoS): measurements of image formats in social cloud computing." *Multimedia Tools and Applications* (2020): 1-26.
- [35] Kumar, V., Laghari, A. A., Karim, S., Shakir,

- M., & Brohi, A. A. (2019). Comparison of fog computing & cloud computing. *Int. J. Math. Sci. Comput*, 1, 31-41.
- [36] Halepoto, I. A., Memon, M. S., & Parveen, S. (2017). Analysis of quality of experience frameworks for cloud computing. *IJCSNS*, 17(12), 228.
- [37] Shafiq, M., & Khan, A. (2018). Assessment of quality of experience (QoE) of image compression in social cloud computing. *Multiagent and Grid Systems*, 14(2), 125-143.
- [38] Laghari, A. A., He, H., Karim, S., Shah, H. A., & Karn, N.K. (2017). Quality of experience assessment of video quality in social clouds. *Wireless Communications and Mobile Computing*, 2017.
- [39] Laghari, A. A., He, H., Shafiq, M., & Khan, A. (2017, May). Impact of storage of mobile on quality of experience (QoE) at user level accessing cloud. In *2017 IEEE 9th international conference on communication software and networks (ICCSN)* (pp. 1402-1409). IEEE.
- [41] Memon, K. A., Halepoto, I. A., & Khan, A. (2019). Quality of experience (QoE) in cloud gaming models: A review. *Multiagent and Grid Systems*, 15(3), 289-304.
- [42] Laghari, Asif Ali, Hui He, Muhammad Shafiq, and Asiya Khan. "Assessing effect of Cloud distance on end user's Quality of Experience (QoE)." In *2016 2nd IEEE international conference on computer and communications (ICCC)*, pp. 500-505. IEEE, 2016.
- [43] Laghari, A., He, H., Laghari, R., Khan, A., & Yadav, R. (2019). Cache Performance Optimization of QoC Framework. *EAI Endorsed Transactions on Scalable Information Systems*, 6(20).
- [44] Wagan, A., & Umrani, A. (2020). Effect of Packet Loss and Reorder on Quality of Audio Streaming. *EAI Endorsed Transactions on Scalable Information Systems*, 7(24).