Journal of Autonomous Intelligence (2024) Volume 7 Issue 5
doi: 10.32629/jai.v7i5.1649

ORIGINAL RESEARCH ARTICLE

An improved algorithm architecture for trust generation in Social Cloud using improved meta-heuristic

Santosh Kumar*, Sandip Kumar Goyal

Department of Computer Science Engineering, MMEC, MM (DU), Mullana, Haryana, Ambala 133-207, India

*** Corresponding author:** Santosh Kumar, santosh.iete@gmail.com

ABSTRACT

In the rapidly evolving landscape of Social Cloud, where online networks leverage real-life social relationships, the assessment of cloud service provider quality hinges on established trust and reputation. This study addresses the crucial factors influencing service quality by delving into multi-user collaboration, resource sharing, and feedback within the Social Cloud. The problem of selection of strong and trustworthy service provider is addressed in this article. Our approach involves a two-fold process. Firstly, we employ a statistical evaluation to generate trust in cloud services. Secondly, optimization strategies are introduced through the application of the artificial bee colony (ABC) algorithm, drawing inspiration from the social group behaviour of honey bees. This innovative methodology aims to enhance the trustworthiness and reliability of deployed cloud services in the Social Cloud environment. To validate our proposed framework, we conduct simulation analyses comparing its performance against existing approaches. The results showcase the effectiveness of our method, which, inspired by ABC as a metaheuristic technique, establishes a trustworthy and reliable foundation for cloud services within the dynamic Social Cloud context. This work contributes to the ongoing discourse on trust evaluation in cloud services, offering a novel perspective and practical insights.

Keywords: Social Cloud; artificial bee colony (ABC); trust generation

ARTICLE INFO

Received: 15 March 2024
Accepted: 2 April 2024
Available online: 10 July 2024

1. Introduction

The word social refers to the aspects of humans that they create around the world that make them her comfortable. Cloud computing has been viewed as a service network for the last couple of years[1] and hence the issue of service provider selection has become an issue to be addressed[2]. A social cloud can be defined as follows.

Definition 1. *A Social Cloud Network (SCN) is a network of 'N' number of users that are oriented with 'S' number of services where* $s \in \{1,2,\dots,S\}, S \neq \infty$ *and* $n \in \{1,2,\dots,N\}, N \neq \infty$. *The network has '*$A_N$*' number of active service provider and each service provider may provider one or more than one service in one go. There are '*S_N*' number of seekers in the network where* $A_N \in s_S$ *as well as* $S_N \in s_S$[3].

The seekers supply the list of services that are required by them and out of the active users, a service provider has to be selected. The problem of service provider selection can be viewed using the following objective function.

$$f = argmax_{\forall S}|\text{QoS}| \quad (1)$$

The objective function is to maximize the Quality of Service (QoS) for every service that is listed in s. The selection of the service provider has to be done on the base of the service experience of the service provider, and the knowledge factor of the service seeker from the service provider[4]. In the light of the above discussion, the problem statement can be defined as follows:

In the context of social cloud computing, choosing the best service providers is a crucial problem. Because the Social Cloud Network (SCN) includes a wide range of users and services, a strong provider selection process is required. In order to maximize Quality of Service (QoS), this method must take into account the knowledge level of seekers as well as the service experience of providers. In order to improve user happiness and network performance, this problem calls for the creation of an enhanced algorithm architecture for trust formation within the social cloud. The purpose of this research study is to suggest an architecture that will further the paradigms of social cloud computing. Various knowledge factors can be associated with a service seeker S_{N_S} as follows in **Figure 1** as follows.

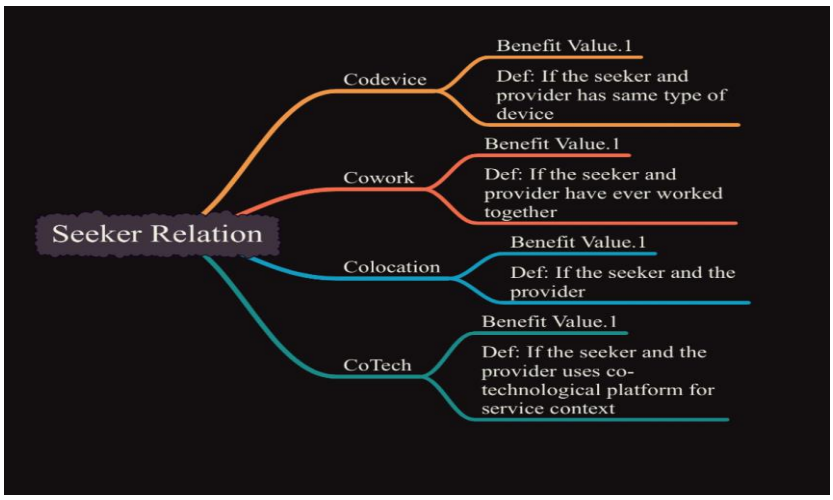


Figure 1. The relation factors.

Figure demonstrates that a seeker can be associated with four types of relations with the provider[5] as follows:

1) Co-device: If both the seeker and the provider are using the same type of device. The factor is valuable as the inter-dependability between the same type of devices has been observed to be better.
2) Co-work: Co-work is one of the most exclusive parameters for the analysis of the relation factor. Working together creates trust in each other and vice versa.
3) Co-location: Another valid parameter that establishes the knowingness of two people in a given group. If two people reside nearby, the colocation factor is utilized.
4) Co-Tech: It is a technology-dependent parameter which illustrates that if the seeker and the provider use the same technological platform, they will result in better performance efficiency.

All these factors contribute to building the trust of a human for job orientation. The concept of relationships does not provide any guarantee of job satisfaction and hence Quality of Service (QoS) becomes crucial at the time of the selection of the service provider[6]. There are several aspects of the selection of a provider even if the relationships are not considered[7].

Motivation statement: The driving force behind work is tackling the urgent need to improve the establishment of trust in the ever-changing social cloud environments. The paper presents a sophisticated algorithm design in recognition of the critical function that relationships and past provider records based on QoS factors play. Enhancing trust evaluations through the use of enhanced meta-heuristic and statistical

techniques is the main goal of the article. The suggested method presents a novel fitness function and grouping behaviour of artificial bees by utilizing a modified ABC optimization technique. Practical application and real-world relevance are ensured by the focus on the S-IoT dataset. In the end, the research aims to support the creation of more accurate and dependable trust-generating processes in social cloud computing, promoting a safe and dependable environment for both service providers and consumers.

Further, research article uses a simulation model to empower the service provider selection policy utilizing the QoS parameters. Swarm Intelligence (SI) based algorithmic architectures have been proposed and have been utilized in the selection of the service provider in the case of Social Cloud Network[8,9]. Artificial bee colony (ABC), cuckoo search and firefly algorithms have been modified time by time for the service orientation for the last couple of years[8,10]. Though, trust evaluation using QoS parameters has been also utilized earlier the main contribution of the paper is as follows:

1) Division of the service context into three categories for the trust generation.
2) Development of a novel behavior based on a meta-heuristic approach.
3) Tuning and development of deep neural network for precise training and validation of the generated context categories.
4) Generation of rank using classified architecture.

The rest of the paper is organized in the following manner.

The second section illustrates the related work that includes trust generation through statistical approaches and SI algorithm architecture as well. The proposed work is illustrated in section 3 whereas the results and discussion have been presented in section 4. The paper is concluded in section 5.

2. Related work

The cloud environment has played as a collaborator in the recent past and to add to its strengths. More diversification and a multitude of concepts have been employed to cultivate trust within the social cloud. A comprehensive survey addressing this was conducted by Caton et al.[11], delving into the foundational aspects of trust within social clouds. However, while the survey provides valuable insights, it lacks an in-depth exploration of emerging trust mechanisms and their applicability in dynamic cloud environments. Further research is warranted to address these gaps and enhance understanding of trust dynamics within social cloud systems[11]. The reputation of service providers and the trust vested by clients are pivotal in the cloud computing market. However, vulnerabilities within services can compromise trust and integrity, leading to a loss of confidence among clients and users. While Macias and Guitart[12] offer a trust model analysis, it primarily focuses on statistical analysis of feedback reports and lacks mechanisms to address emerging threats and ensure robust trust enforcement. There's a need for novel approaches that incorporate dynamic trust mechanisms and proactive risk mitigation strategies to bolster trust in cloud environments[12]. Yan et al.[13] proposed a scheme for secure access to cloud data based on trust evaluation. While their approach leverages reputation and user trust in service providers, it primarily relies on attribute-based encryption and proxy re-encryption techniques. However, the scheme may overlook emerging threats and evolving trust dynamics. Future research should explore hybrid trust models integrating diverse trust mechanisms to enhance security and trustworthiness in cloud data access[13]. In the realm of providing accurate and trustworthy web services, Wang et al. introduced a Quality of Service (QoS) evaluation framework that considers past track records and user experience. While their approach integrates fuzzy hierarchy and rough set theory for automatic weight calculation, it may not adequately address the evolving nature of user trust and dynamic internet environments. Future endeavours should focus on adaptive QoS evaluation frameworks that dynamically adjust to changing trust dynamics and user expectations[14]. Zanbouri and Navimipour[15] discussed optimization strategies inspired by honey bee behaviour to address service reliability challenges in dynamic environments. While their approach shows promise, it may lack scalability and robustness in highly

dynamic cloud ecosystems. Further research should explore hybrid optimization techniques that combine nature-inspired algorithms with machine-learning approaches to enhance trust-based clustering and service reliability[15]. Lee and Brink[16] presented a framework evaluating end-user trust in adopting software as a service (SaaS) models. While their statistical evaluation highlights factors influencing end-user trust, it may overlook emerging privacy and security concerns inherent in cloud-based services. Future research should focus on comprehensive trust assessment frameworks that integrate multidimensional factors and proactive risk management strategies to enhance end-user trust and confidence in cloud services[16]. Kumar and Tripathi[17] proposed a state-of-the-art framework aimed at enhancing security, privacy, and trust within Industrial Internet of Things (IIoT) applications. Leveraging a comprehensive blockchain-based methodology, the authors addressed critical issues concerning reliability and privacy protection in industrial settings. Their framework offers a robust and secure solution, capitalizing on the advantages of blockchain technology to establish a solid foundation for privacy and trust within IIoT systems. This research significantly contributes to the evolution of industrial network security infrastructure, effectively catering to the evolving demands of the networked and data-driven industrial environment[17]. In recent years, Swarm Intelligence (SI) has gained prominence alongside statistical concepts for evaluating cloud trustworthiness. Kumar and Goyal[18] utilized artificial bee colony (ABC), among various metaheuristics, to optimize service provider selection. Their enhanced ABC approach showcased reliable and successful communication compared to existing solutions. Furthermore, the study evaluated success rates achieved through ABC, Cuckoo search, and Firefly algorithms, demonstrating the efficacy of their proposed methodology[19]. Bangui et al.'s[20] study concentrated on integrating moral AI concepts into Social Internet of Things (SIoT) trust management. While the study addresses ethical issues, it lacks a thorough analysis of the usefulness of the suggested guidelines, thereby revealing certain shortcomings. More empirical data and real-world application scenarios are needed to confirm the efficacy of ethical AI principles in actual SIoT systems. This highlights the necessity for further research to bridge this gap and provide a more comprehensive understanding of the practical implications of moral AI concepts in SIoT trust management[20]. Similarly, Ouechtati et al.[21] proposed a fuzzy logic-based model to filter fraudulent recommendations in the Internet of Things (IoT). While their approach leverages fuzzy logic for precision, it may be vulnerable to varying levels of uncertainty. Furthermore, the absence of comprehensive validation across multiple datasets limits the model's applicability. To address these limitations, further research is needed to assess the model's performance under dynamic and changing SIoT circumstances. This underscores the importance of ongoing investigation to enhance the robustness and reliability of fraud detection mechanisms in IoT environments[21]. In another study, Mohana et al. introduced an AI-enabled simulator for categorization, grouping, and navigation in the SIoT. However, a notable drawback is the lack of in-depth analysis of the simulator's functionality across various SIoT scenarios. To validate the superiority of their approach, a more thorough comparison with existing simulators and real-world implementations is warranted. Additionally, addressing potential biases is essential to ensure the credibility and impartiality of simulation results. Thus, future studies should focus on comprehensive evaluation and validation to enhance the effectiveness and applicability of AI-driven simulators in the dynamic SIoT landscape[22].

3. Proposed work

As it has been illustrated trust generation is a dependent factor of the relationships that the seeker has with the provider and the record of the provider based on the QoS parameters. Keeping this in mind, the proposed work is divided into two sections namely the trust generation using statistical approaches followed by the amendments that have been made in the SI algorithm architecture. For the proposed case scenario, the ABC algorithm has been modified which includes a new novel fitness function along with a new grouping behaviour of the artificial bees. The proposed work can be illustrated using the following flow diagram.

In the first case scenario, the proposed work starts with the simulation work where the dataset is loaded. The dataset has been considered from the social IoT dataset[23] that is available as open-source content for researchers and other types of analysis in the world. The dataset contains the service records and type of services that are offered by the providers. Further, **Table 1** provides a work illustration number of services offered by six types of services offered by 16 types of devices in the network.

Table 1. Work Illustration.

Device type	ID of service 1	ID of service 2	ID of service 3	ID of service 4	ID of service 5	ID of service 6
1	1	2	4	5	7	10
2	1	2	4	5	7	8
3	1	2	4	5	7	10
4	15					
5	2	4	7			
6	2	4	7			
7	7	16				
8	5	7	7			
9	1	2	3	4		
10	1	2	5	6	7	
11	1	2	6	7	8	9
12	1	2	3	5	7	8
13	1	2	7	13	8	
14	1	2	7	9		
15	1	2	7	14		
16	1	2	4	7	8	

The proposed work runs a simulation architecture that initializes a service requirement and it is broadcasted in the S-IoT architecture. As shown in **Figure 2**, the overall work process is divided into 15 subsequent steps. The proposed work discusses two kinds of feedback in the list namely short feedback (S_f) and regular feedback (R_f). Both the feedbacks are defined using Equations (2) and (3) as follows.

$$S_f = \sum_{i=1}^{S} \sum_{j=1}^{N} fd_{ij} \tag{2}$$

where *fd* is the feedback ranging from 1–5 and S_f contains all the feedback that has been received by N number of total users for S number of services. R_f is the mean of all feedback against each service. Hence if there are S number of services, R_f is an array which contains $1 \times S$ number of identities.

$$R_f = \frac{\sum_{i=1}^{S} S_{f_i}}{S} \tag{3}$$

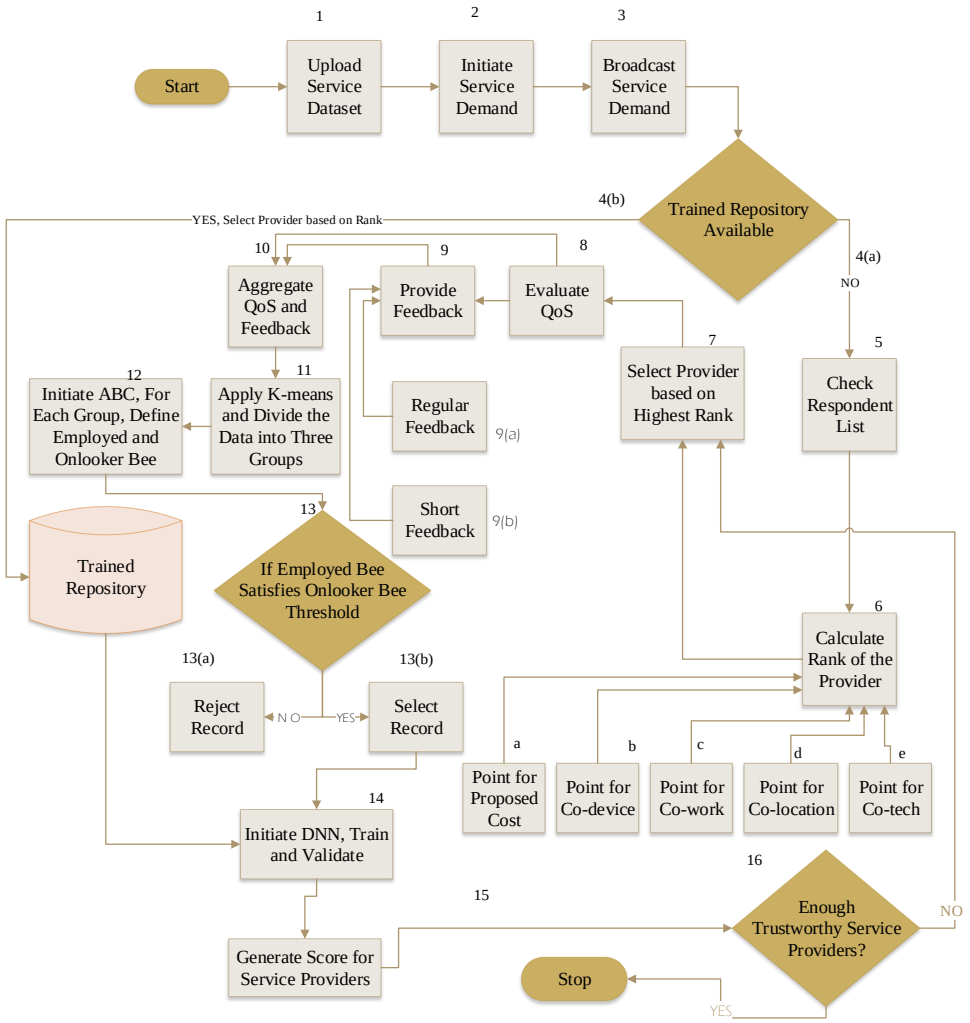


Figure 2. The proposed work.

The collected feedback is now divided into 3 subsequent groups using k-means[24]. The purpose is to divide the providers on an overall basis so that they can be further recommended for any service. When the data is divided into three groups, each user will fall into multiple groups as one service provider has the capability to provide *S* number of services. The proposed work develops a learning method that can be utilised to rank the users based on their S_f and R_f against the services. Hence to train the system, the logic is to select the best suitable records that match the group sequence. For this purpose, ABC has been applied due to its significant evidence submitted in the literature survey[8,15]. The algorithmic architecture of the ABC algorithm is provided as follows. The ABC algorithm architecture is made up of three kinds of bees namely the employed bee, the onlooker bee and the scout bee. The employed bee is the bee that goes in search of food and gathers the juices from the flowers as nectars. The onlooker bees are the bees that judge the nectar and whether it will be accepted or not. Once the employed bee is tired of providing services of nectar, it will be difficult for the employed bee to produce high-quality nectar and require rest. Considering this in mind, the onlooker bee provides rest to the employed bee by rejecting the food of the employed bee. Once all the employed bees are either scout bees or if the total desired food is attained, the food search process is terminated. In the case of the proposed work, the employed bee is the service provider that has been categorized into any group based on the k-means clustering algorithm. The general flow chart for the ABC algorithm is shown in **Figure 3** as follows.

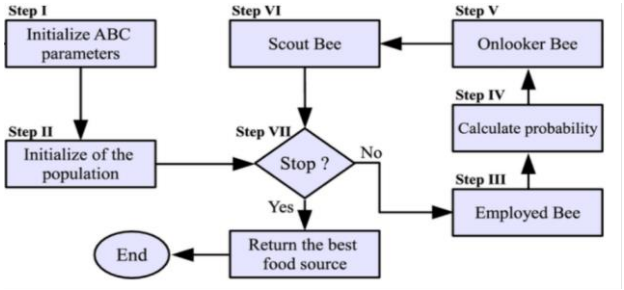


Figure 3. General flow chart of ABC algorithm architecture.

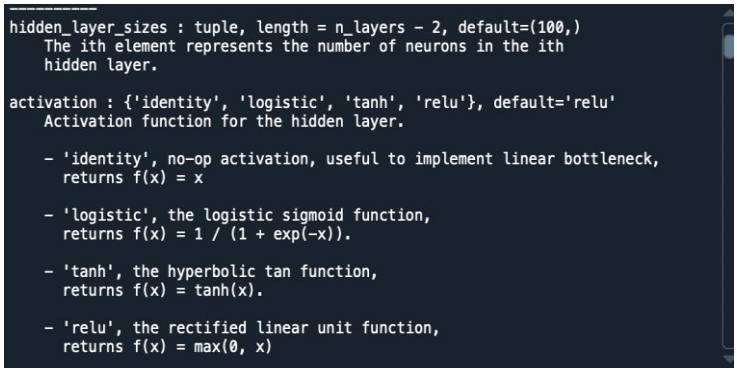
An enhanced algorithm architecture for trust generation in the social cloud utilizing an upgraded meta-heuristic approach is described along with the "Grouped-ABC" method. The method uses an adapted version of the artificial bee colony (ABC) optimization technique to improve the process of building trust in a social cloud setting. A feature vector containing social and reliability traits (S_f, R_f) makes up the input. The approach starts by utilizing k-means clustering to group the feature vector, forming unique groups or 'hives' for the purpose of evaluating trust later on. Furthermore, the clustered data is labelled using statistical machine learning, which improves the accuracy of trust evaluations reflecting a novel implication of proposed Grouped-ABC. The proposed ABC algorithm can be illustrated using the following algorithm sequence.

Algorithm 1 Algorithm Grouped-ABC

```
Input: Feature Vector [S_f , R_f] as Fv
[k_ind, k_cent] = kmean(Fv, tbh); //tbh is total bee hives which is 3 in case of proposed work
Apply Statistical Machine Learning to label the k_ind
Establish Ground_Truth as GT = [Good, Moderate, Bad];
for i in GT.unique() // Fetch unique classes from GT
   Hivebees = Find(GT == i) // Fetch the elements of current hive
   Hivefood = Fv[Hivebees];
   Hiveglobal = kcent.i;
  for j in Hivebees // Considering each bee once
      levyflight = 10; // Generate a Levy Flight with random directions
     flightreward = [] // Initialize a flight reward array to empty
     for f = 1:levyflight
        Grouporder = Random(Hivebees) // Generate a random population
        Foodtoevaluate = Fv.Grouporder
        Onlookerjudgement = d1 = Mean(Foodtoevaluate + Hiveglobal);
        d2 = Eucl (d1, Hiveglobal); // Euclidean distance between current group hives and global hive
        d3 = Eucl(Hivefood, Hiveglobal) // evaluate d1, d2 and d3 for local and global best
        dx = d2 - d3;
        dx = dx / d3
        if dx * 100 < 35reward[If] = 10; // assign 10 reward points for flight.  else
          reward[If] = 0; // assign penalty or 0 reward
          if the local food and global food is less than 35 percent different from each other
       EndIf
    EndForf
        If rs = SUM(reward);
          St = 50; // selection threshold
          if rs ≥ St; // if reward is more than 50 percent for all flights, accept record
      else
          reject record
EndForj
EndFori
```

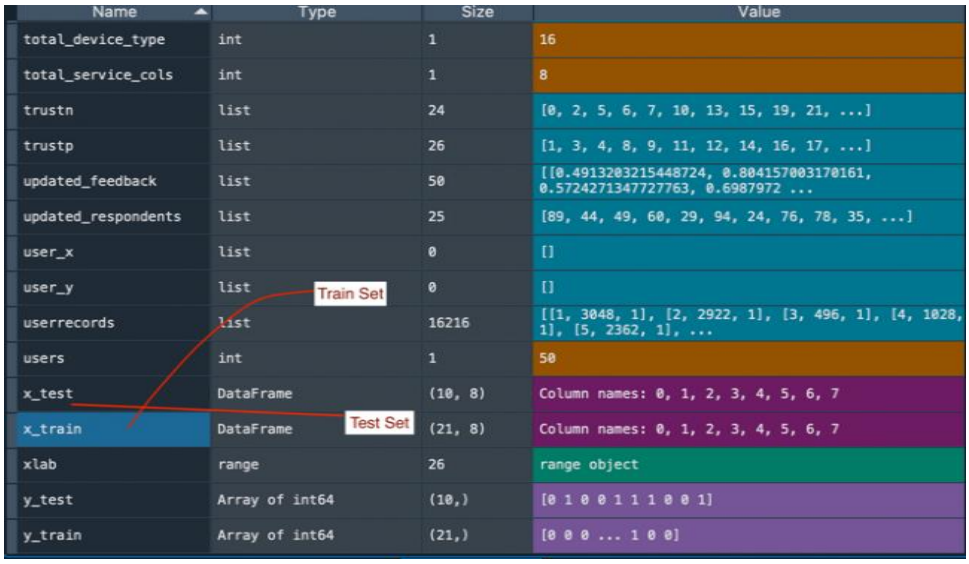
The method makes use of a ground truth architecture that includes classes like “Good,” “Moderate,” and “Bad,” which guarantees a thorough assessment of reliability. The next iterative step simulates a bee colony by initializing hives and individual objects within them. To mimic bees’ exploratory instincts and randomness, a Levy flight mechanism is implemented. The random population group is assessed for every flight, and an observer’s assessment is derived from the Euclidean distance between the group hives in question and a global hive. Based on a variety of factors, such as the distinction between local and global food sources, rewards are allocated to aircraft. Based on the total reward, the algorithm establishes a selection threshold and decides which records to accept or reject. Interestingly, the acceptance criterion takes into account the portion of reward points earned throughout all flights. By combining clustering, labelling, and meta-heuristic optimization, this enhanced meta-heuristic technique demonstrates an advanced trust generation system in a social cloud setting, improving the precision and efficacy of trust assessments.

The selected records from the proposed ABC algorithm are passed to a deep neural network from training and classification. More true classified records refer to high rank in the system and are recommended for the service. In order to train and classify, the ordinal measures that are illustrated in the table have been utilized. The proposed work has been implemented in Python and the development process is presented in **Figures 4–6**.



```
hidden_layer_sizes : tuple, length = n_layers - 2, default=(100,)
    The ith element represents the number of neurons in the ith
    hidden layer.

activation : {'identity', 'logistic', 'tanh', 'relu'}, default='relu'
    Activation function for the hidden layer.

    - 'identity', no-op activation, useful to implement linear bottleneck,
      returns f(x) = x

    - 'logistic', the logistic sigmoid function,
      returns f(x) = 1 / (1 + exp(-x)).

    - 'tanh', the hyperbolic tan function,
      returns f(x) = tanh(x).

    - 'relu', the rectified linear unit function,
      returns f(x) = max(0, x)
```

Figure 4. Neural network ordinal activations.



Name	Type	Size	Value
total_device_type	int	1	16
total_service_cols	int	1	8
trustn	list	24	[0, 2, 5, 6, 7, 10, 13, 15, 19, 21, ...]
trustp	list	26	[1, 3, 4, 8, 9, 11, 12, 14, 16, 17, ...]
updated_feedback	list	50	[[0.4913203215448724, 0.804157003170161, 0.5724271347727763, 0.6987972 ...
updated_respondents	list	25	[89, 44, 49, 60, 29, 94, 24, 76, 78, 35, ...]
user_x	list	0	[]
user_y	list	0	[]
userrecords	list	16216	[[1, 3048, 1], [2, 2922, 1], [3, 496, 1], [4, 1028, 1], [5, 2362, 1], ...
users	int	1	50
x_test	DataFrame	(10, 8)	Column names: 0, 1, 2, 3, 4, 5, 6, 7
x_train	DataFrame	(21, 8)	Column names: 0, 1, 2, 3, 4, 5, 6, 7
xlab	range	26	range object
y_test	Array of int64	(10,)	[0 1 0 0 1 1 1 0 0 1]
y_train	Array of int64	(21,)	[0 0 0 ... 1 0 0]

Figure 5. Train test architecture for neural.

```
 onlooker
for ia in myrange:
    pairingbee=0;
    pairingbee=round(last_value*random.random());
    if pairingbee>len(food_source):
        pairingbee=0;
    try:

        food_effort.append(food_source[pairingbee]);
    except:
        food_effort.append(food_source[0]);
     preparing competion bee values who flew in the simiar directi
    comp_effort=[];
    myrange1=range(0,6);
 pdb.set_trace();
for ia in myrange1:
    comp_index=0;
     pdb.set_trace();
    comp_index=round(last_value*random.random());
    if comp_index>len(food_source):
        comp_index=0;
    try:

        comp_effort.append(food_source[comp_index]);
    except:
        comp_effort.append(food_source[0])

 pdb.set_trace();
bee_fit_value=bee_fitness(food_effort,comp_effort);

if bee_fit_value>.5:
    try:
         pdb.set_trace();
        bee_fit_index.append(orgindex);
        try:

            sm=food_source[orgindex];
        except:
```

Figure 6. Application of proposed ABC algorithm.

The ordinal measures of the neural network used in the proposed work are given in **Table 2**.

Table 2. Ordinal measures.

Parameter	Description
Total number of layers	5–10
Total number of instances at the input layer	55,000
Activation function	Logistic sigmoid
Hyperbolic function	Tan*H* hyperbolic
Rely	Linear

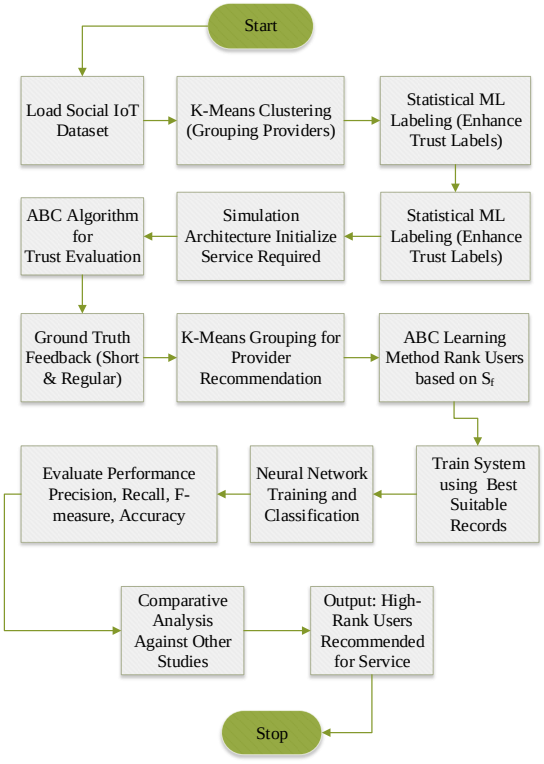


Figure 7. System process flow of proposed methodology.

It should be noted here that a high classification rate indicates a high rank in the system and is recommended for the job in future. The overall system process flow of the proposed methodology is summarized to in **Figure 7** to present a simplified and detailed illustration of the work methodology. The evaluation of the work architecture is detailed in the next section.

4. Results and discussion

The performance of the proposed ABC-inspired learning mechanism is evaluated against existing studies to justify the effectiveness of the proposed work. The number of test samples used in the simulation analysis varied from 1000 to 30,000. The performance metrics used for the evaluation are precision, recall, f-measure and accuracy. The comparative analysis of the proposed work against multi-SVM, Parveen et al.[18] and Marudhadevi et al.[4] work is summarized in **Table 3**. It is observed that with an increase in the number of test samples the corresponding precision of all the studies also increased. It is noteworthy that all models' precision values appear to vary with varying sample sizes. On the other hand, several models exhibit more steady performance patterns than others. In comparison to the other models, the suggested model and the Marudhadevi et al.[4] model, for example, show somewhat stable performance patterns across a range of sample sizes. Although precision is a crucial indicator, a thorough assessment of these models would also look at recall and F1-score, among other performance metrics. However, the precision of the proposed work remained highest with an average value of 0.89312616, followed by multiSVM of 0.794920183, Parveen et al.[18] of 0.814634929 and Marudhadevi et al.[4] of 0.80056456. This comparatively higher precision of the proposed work is mainly due to the designed learning mechanism with the integration of the ABC fitness function.

Table 3. Precision values.

Total test samples	Proposed	multiSVM	Parveen et al.[18]	Marudhadevi et al.[4]
1000	0.85597133	0.73152343	0.81332147	0.7866021
2000	0.86958376	0.828472	0.82940539	0.85154697
3000	0.86111022	0.76135469	0.79790861	0.71572289
4000	0.86080953	0.85186259	0.86062009	0.85792603
5000	0.85705448	0.76933413	0.80045128	0.70334992
6000	0.88277944	0.73214908	0.78039564	0.82329001
7000	0.87614795	0.85030415	0.72476549	0.72182522
8000	0.8932013	0.76596086	0.84167887	0.7710697
9000	0.86494587	0.71310537	0.76532151	0.74318359
10,000	0.88607585	0.81026058	0.80424203	0.85513865
11,000	0.86038045	0.85610483	0.72587394	0.71321543
12,000	0.90072651	0.7717469	0.75455164	0.87560023
13,000	0.85364568	0.70396033	0.70478925	0.73735858
14,000	0.89667617	0.8648737	0.76616453	0.81928404
15,000	0.90166823	0.8235654	0.75341953	0.79000456
16,000	0.92882244	0.9075303	0.9031493	0.82090117
17,000	0.93561456	0.80338035	0.86410536	0.80905449
18,000	0.87070171	0.74515324	0.84333187	0.76665456
19,000	0.89150509	0.74116263	0.87052687	0.83209188
20,000	0.86908435	0.75471032	0.80551467	0.77984431
21,000	0.92615446	0.8326628	0.85772972	0.87334547

Table 3. (*Continued*).

Total test samples	Proposed	multiSVM	Parveen et al.[18]	Marudhadevi et al.[4]
22,000	0.88555885	0.7458797	0.78545496	0.80454962
23,000	0.97109774	0.79811516	0.86412558	0.84465413
24,000	0.88882271	0.84935448	0.8512458	0.7471869
25,000	0.87205607	0.73004272	0.84528689	0.78719496
26,000	0.87267952	0.73195781	0.84643795	0.85590282
27,000	0.89919665	0.77605159	0.80417499	0.80623768
28,000	0.93039162	0.83810911	0.81478078	0.78241281
29,000	0.97376159	0.86084122	0.93056903	0.82864257
30,000	0.95756068	0.89807602	0.82970484	0.91314552

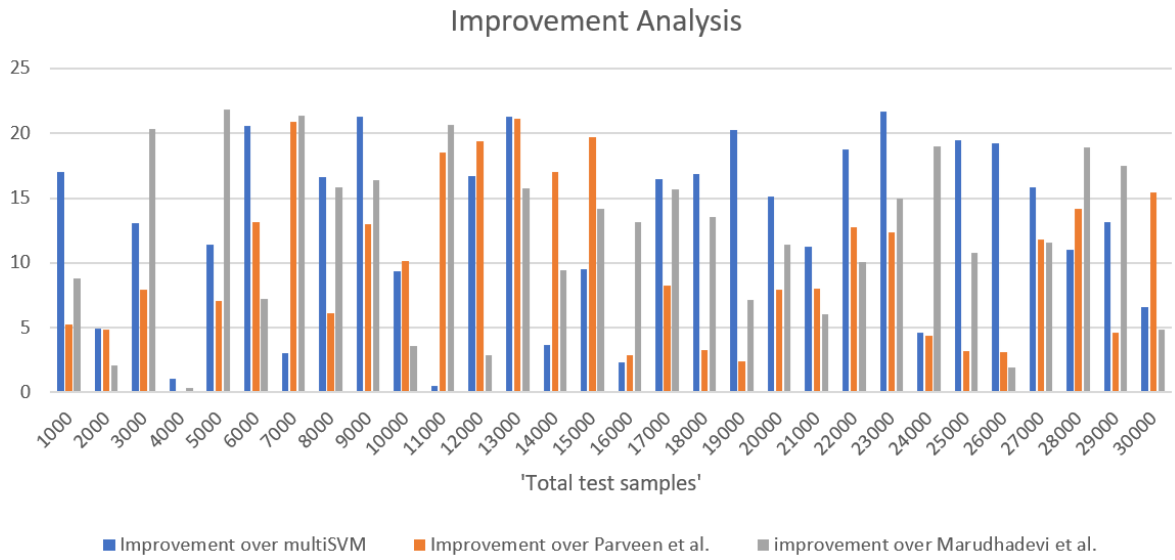


Figure 8. Precision improvement analysis.

The suggested model's efficacy in comparison to other cited models in the literature is demonstrated by the **Table 3**, which offers insightful information about the accuracy performance of various models across a range of sample sizes. The tabulated precision values further show that the proposed work outperformed the existing works. Insightful information about the performance comparison between the suggested technique and current methods may be gained from the study shown in **Figure 8**. It clearly shows that the suggested method offers notable improvements above accepted standards. The results indicate that the suggested model outperforms the multiSVM technique in terms of prediction accuracy, with an average improvement of 12.75%. In a similar vein, the suggested methodology exhibits a notable improvement of 9.95% in comparison to the model presented by Parveen et al.[18], demonstrating its efficacy in resolving current restrictions. Moreover, the suggested method shows a notable average improvement of 11.90% compared to the model suggested by Marudhadevi et al.[4], highlighting its ability to produce better performance results. These results highlight the effectiveness of the suggested methodology and demonstrate its potential to surpass well-established methodologies in the field, which will further propel the development of predictive modeling frameworks.

Recall analysis given in **Table 4** provides a thorough summary of how well the suggested approach and previous research performed with different amounts of test samples used in the simulation analysis. After analysis, it is clear that the suggested methodology continuously maintains a high average recall of 0.873084603 over the span of 1000 to 30,000 test samples. Comparatively, over the same range of test samples, multiSVM, Parveen et al.[18], and Marudhadevi et al.[4] show average recall values of 0.791554442,

0.790272661, and 0.794430241, respectively. This comparative study highlights how well the suggested methodology performs in terms of recall values, consistently outperforming the previous research for all test sample sizes taken into account. These results demonstrate the robustness and efficacy of the suggested method in precisely locating pertinent instances within the dataset, underscoring its potential for useful implementations in real-world settings.

Table 4. Recall value.

Total test samples	Proposed	multiSVM	Parveen et al.[18]	Marudhadevi et al.[4]
1000	0.843504	0.7092106	0.8303771	0.7816142
2000	0.84412	0.7453756	0.7721485	0.7243327
3000	0.841291	0.7165138	0.8088363	0.8030737
4000	0.84072	0.6975408	0.8020651	0.794051
5000	0.85727	0.8158593	0.7287135	0.7420956
6000	0.869675	0.7143964	0.7551031	0.802839
7000	0.864387	0.8590797	0.7915564	0.7943809
8000	0.857317	0.818219	0.7609411	0.7275091
9000	0.862495	0.7811742	0.8035194	0.7744287
10,000	0.859868	0.812133	0.7189369	0.8539949
11,000	0.865695	0.7772566	0.8574158	0.8221095
12,000	0.851045	0.7560323	0.7684519	0.8257217
13,000	0.842732	0.74217	0.7266247	0.8129945
14,000	0.898361	0.8359569	0.8219659	0.871712
15000	0.844738	0.7254166	0.7875228	0.8327544
16,000	0.878934	0.8368725	0.7853537	0.7278589
17,000	0.866485	0.7288112	0.7287021	0.7846366
18,000	0.861618	0.7810061	0.7157842	0.7186202
19,000	0.903625	0.8466999	0.878797	0.8426491
20,000	0.85367	0.8288511	0.7437844	0.7792445
21,000	0.943512	0.9001102	0.8655688	0.809819
22,000	0.864556	0.7788825	0.7548099	0.715743
23,000	0.880735	0.8136469	0.8060022	0.8562753
24,000	0.94944	0.8356119	0.9052978	0.7820246
25,000	0.924666	0.8561124	0.7690039	0.7888677
26,000	0.92878	0.8101874	0.8606652	0.7854365
27,000	0.84583	0.7504803	0.7995077	0.713256
28,000	0.905786	0.827535	0.7942193	0.877652
29,000	0.851225	0.802681	0.8135516	0.8366021
30,000	0.890459	0.8428101	0.7529537	0.8506101

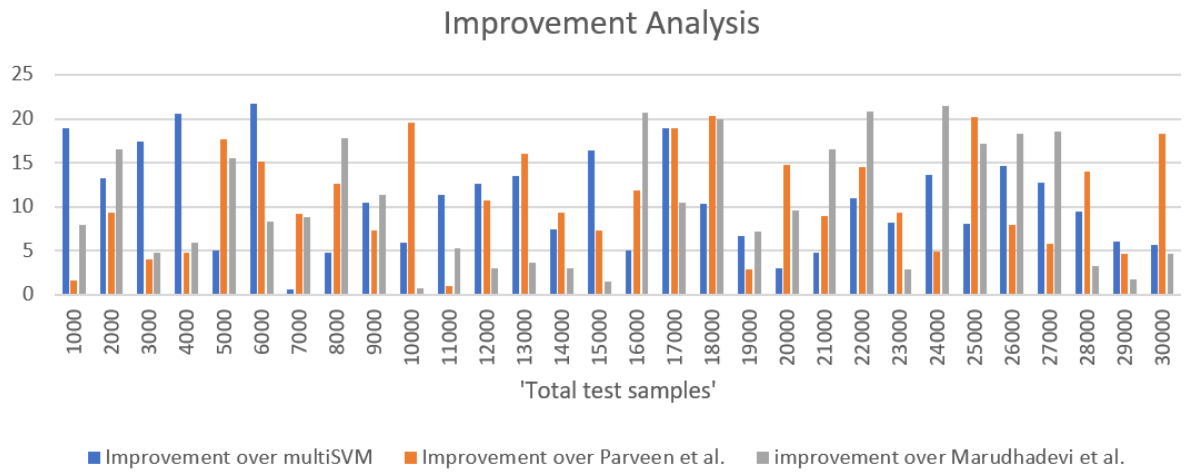


Figure 9. Recall improvement analysis.

The improvement analysis for recall values, as shown in **Figure 9**, is highlighted in the discussion. It clarifies that the suggested methodology exhibits significant improvements over current methods. In particular, the suggested approach shows average improvements of 10.60%, 10.76%, and 10.24%, respectively, in comparison to multi-SVM, Parveen et al.[18], and Marudhadevi et al.[4] This shows that the suggested methodology consistently outperforms other approaches using a range of comparable indicators. The efficiency of the suggested strategy is demonstrated by these noteworthy improvements, which can be attributed to its implementation of a learning system for user rating based on service feedback. Through the utilization of this mechanism, the suggested methodology improves recall values by more precisely identifying and prioritizing pertinent instances. Thus, these results support the idea that the suggested methodology is unique. These results thus support the idea that the suggested technique is the best in class, indicating that it has the capacity to provide dependable and effective solutions in the field of service feedback analysis.

Similarly, **Table 5** summarizes the comparative analysis performed in terms of f-measure values. F-measure represents the harmonic mean of precision and recall values observed for the simulation analysis. Thus, the proposed work exhibited a comparatively higher f-measure of 0.882561522 computed against test samples ranging from 1000 to 30,000. The average f-measure observed for multiSVM is 0.791915646, Parveen et al.[18] is 0.800994978, and Marudhadevi et al.[4] is 0.796288023.

Table 5. F-measure values.

Total test samples	Proposed	multiSVM	Parveen et al.[18]	Marudhadevi et al.[4]
1000	0.849692	0.720194	0.821761	0.7841
2000	0.856663	0.78473	0.799753	0.782805
3000	0.851085	0.738254	0.803335	0.756886
4000	0.850646	0.767016	0.830312	0.824754
5000	0.857162	0.791914	0.7629	0.722203
6000	0.876178	0.723164	0.767541	0.812936
7000	0.870228	0.854669	0.75669	0.756367
8000	0.874891	0.791228	0.799276	0.748656
9000	0.863718	0.745589	0.783955	0.758485
10,000	0.872775	0.811196	0.759201	0.854566
11,000	0.863029	0.814778	0.786181	0.763801
12,000	0.875181	0.763809	0.761438	0.84993

Table 5. (*Continued*).

Total test samples	Proposed	multiSVM	Parveen et al.[18]	Marudhadevi et al.[4]
13,000	0.848154	0.72256	0.71554	0.773332
14,000	0.897518	0.850169	0.793085	0.844685
15,000	0.872275	0.771381	0.770094	0.810816
16,000	0.90319	0.87077	0.840143	0.771585
17,000	0.899724	0.764281	0.790648	0.796659
18,000	0.866136	0.762659	0.774341	0.741861
19,000	0.897524	0.790424	0.874642	0.837337
20,000	0.861308	0.790045	0.77342	0.779544
21,000	0.934753	0.865074	0.861631	0.840383
22,000	0.874931	0.762024	0.769828	0.757553
23,000	0.923712	0.805806	0.834053	0.850425
24,000	0.918132	0.842427	0.87744	0.764209
25,000	0.897591	0.788067	0.805343	0.78803
26,000	0.899856	0.769088	0.853492	0.819157
27,000	0.871697	0.763052	0.801835	0.756902
28,000	0.917924	0.832789	0.804369	0.8273
29,000	0.908379	0.830744	0.868135	0.832603
30,000	0.922792	0.869566	0.789468	0.880769

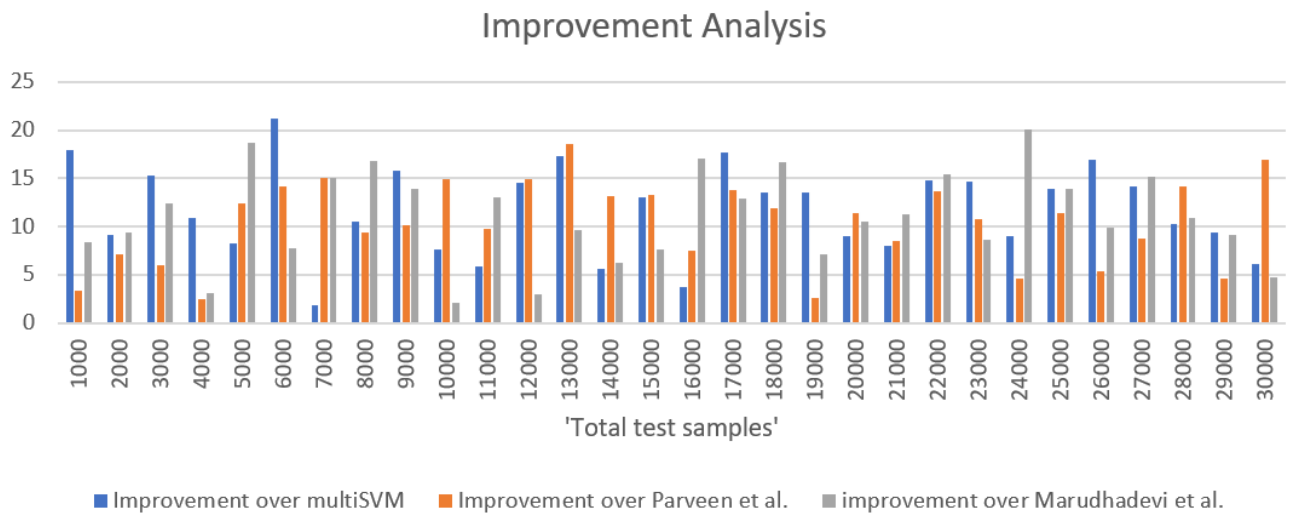


Figure 10. F-measure improvement analysis.

Further, a detailed improvement analysis for f-measure values for the proposed work over the existing studies is shown in **Figure 10**. It is observed that an average improvement of 11.66% is observed for the proposed work over multiSVM, 10.35% over Parveen et al.[18] and 11.02% over Marudhadevi et al.[4] work. It is concluded that the feedback-based selection strategy formed the basis of the enhanced performance observed for the proposed work.

The accuracy of performed service selection to deliver trustworthy and successful communications holds great significance in the presented research. Hence, comparative analysis in terms of accuracy is also performed in addition to precision, recall and f-measure analysis. The accuracy of the classification of the service providers to offer reliable and trustworthy cloud service for all the studies is summarized in **Table 6**. The tabulated values show that the proposed work exhibited the highest accuracy among all the studies with

an average value of 92.66%. The average accuracy obtained using multiSVM is 83.69%, Parveen et al.[18] is 84.43%, and using Marudhadevi et al.[4] is 84.99%. The detailed improvement analysis for the comparative analysis is shown in **Figure 11**.

Table 6. Accuracy values.

Total test samples	Proposed	multiSVM	Parveen et al.[18]	Marudhadevi et al.[4]
1000	85.34715	72.264517	81.430915	81.255461
2000	92.52219	82.437711	87.218841	79.712738
3000	91.93439	89.664857	88.342015	79.823061
4000	90.25889	88.404781	83.021764	87.01056
5000	93.49391	83.81243	86.317797	79.224591
6000	92.59434	78.187276	80.600902	87.725075
7000	95.50571	84.014735	89.282673	87.268028
8000	91.83766	81.884526	77.432607	76.79542
9000	92.6902	82.789741	92.551055	82.862136
10,000	90.97372	84.439502	84.237822	90.276682
11,000	86.43267	74.089059	71.938043	72.436683
12,000	93.18146	78.108033	79.885826	84.611871
13,000	89.42695	89.267861	83.0468	80.445385
14,000	89.88893	79.946401	83.267577	87.705949
15,000	90.31982	88.601991	81.770912	82.228577
16,000	91.56353	87.154114	85.20775	77.844721
17,000	92.27192	85.729034	83.931464	85.575005
18,000	88.09717	74.713017	83.190405	85.402091
19,000	92.55558	77.596818	88.665518	89.285039
20,000	90.35988	78.656304	79.939651	75.721331
21,000	95.74315	80.243705	83.538975	84.309261
22,000	89.16034	80.073186	86.35193	88.287098
23,000	97.18794	85.389888	80.88496	89.129366
24,000	99.73443	95.683207	89.0854	86.361114
25,000	98.66835	93.614813	91.471111	96.708307
26,000	96.22515	95.633505	93.4035	86.525722
27,000	91.64858	85.592972	85.404597	91.421907
28,000	99.29738	84.124443	83.449223	97.676826
29,000	96.01743	79.790574	88.362868	86.867149
30,000	95.06294	89.074493	79.828294	89.302714

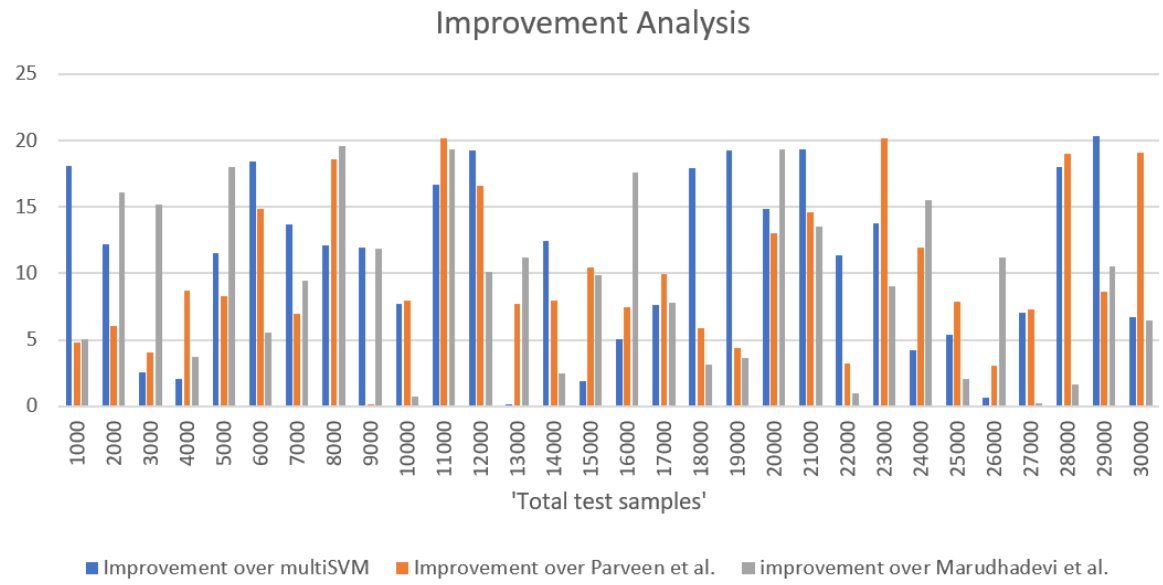


Figure 11. Accuracy improvement analysis.

The improvement exhibited by the proposed work in terms of accuracy values against each variation in the number of test samples is shown in **Figure 11**. It is observed that however the extent of improvement shown by the proposed work varies with change in the number of samples for all the studies, the proposed work outperformed the existing works. The average improvement of 11.08% is exhibited by the proposed work over multiSVM, 9.96% over Parveen et al.[18] and 9.36% over Marudhadevi et al.[4] work. The improved performance of the proposed work is due to a novel learning method developed by the authors under the light of the ranking of the services delivered which was performed under the light of the user's feedback.

The conversation emphasizes the thorough assessment of the suggested methodology by taking into account a number of performance metrics that together show how effective it is. Accuracy values are compared in addition to specific performance criteria like recall, precision, and others being evaluated. The purpose of this analysis is to confirm that the suggested methodology for choosing dependable and strong service providers is appropriate. This comparative analysis is visually represented in **Figure 12**, which compares the performance of the suggested methodology with three previous efforts. The suggested approach's efficacy is verified by comparing accuracy values among these approaches. This comparison analysis is an essential first step toward confirming the superiority of the suggested methodology in terms of making reliable service provider selection easier. As a result, these results bolster the validity and suitability of the suggested methodology in practical situations where precise and knowledgeable choices about service provider selection are critical.

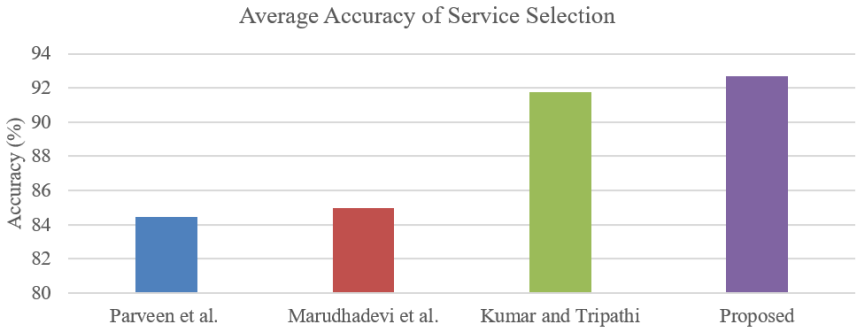


Figure 12. Comparative analysis of average accuracy of service selection for trustworthy service.

The discussion presents architectural elements of the suggested technique, which are intended to

improve connections between providers and seekers by fostering trust. Modifications to the SI algorithm, specifically in the ABC algorithm, and the incorporation of statistical techniques for trust evaluation are fundamental to this architecture. Two notable improvements are the creation of a new fitness function and the introduction of a novel clustering behavior for artificial bees. In contrast to other frameworks such as the DBTP2SF created by Kumar and Tripathi[17], which mainly focus on industrial IoT systems, our approach emphasizes trust dynamics and algorithmic improvements, providing a new angle on enhancing trustworthiness in particular situations. Although the framework proposed by Kumar and Tripathi performs exceptionally well in terms of intrusion detection methodologies, its average accuracy in terms of selecting services that lead to trustworthy communication is 91.745%. This accuracy level is comparable to the findings of Marudhadevi et al.[4] (84.993%) and Parveen et al.[18] (84.435%). The suggested methodology, on the other hand, exhibits an average accuracy that is noticeably higher 92.667%. This significant improvement highlights how well our method works to support trustworthy service selection procedures, highlighting how it may help build trust and improve decision-making in provider-seeker relationships.

The contributions listed in the study support the ways in which the authors enhanced the baseline method in multiple significant ways:

- Division of service context: The authors presented a novel method to thoroughly evaluate trust by breaking the service context into three separate categories for trust generation. This division improves the accuracy and applicability of the trust evaluation process by enabling a more sophisticated understanding of trust dynamics in many circumstances.
- Development of novel behavior: A major improvement over conventional techniques is the introduction of a novel behavior based on a meta-heuristic approach. Using the concepts of meta-heuristic algorithms, this novel methodology maximizes the generation of trust, yielding more effective and efficient results than traditional methods.
- Deep neural network (DNN) tuning and development: Compared to previous approaches, this method represents a significant improvement. A deep neural network that is specifically designed for training and testing the generated context categories has been developed. The authors were able to obtain exact and accurate assessments of trust within each context category by utilizing DNNs, which produced more dependable and robust results.
- Generation of rank via classified architecture: Ranking service providers according to their reliability takes on a more organized and methodical approach when rank is generated via a classified architecture. This architecture creates a more complex and refined ranking mechanism that more accurately captures the subtleties by combining the insights from the division of services context with the use of unique behaviors.

In conclusion, the authors' contributions considerably improve the trust evaluation process by adding novel ideas and methodologies, in addition to addressing the baseline method's current weaknesses. By offering assessments of trust in provider-seeker relationships that are more thorough, accurate, and contextually relevant, these contributions all support the basic premise of enhancing the baseline technique.

5. Conclusion

Cloud computing has shown tremendous popularity and supports numerous online web and IoT applications. However, due to instances of online breaches and compromised communication environments, the reliability of communication remained a challenge. In this respect, the paper addresses the challenge of delivering trustworthy and reliable communication in the social cloud. The paper has developed a learning method that helps in identifying the cloud service providers that offer the least compromised and highly reliable service. To support this, ABC-based optimization algorithm is integrated in addition to ranking performed based on the feedback. The records selected using ABC are passed to a deep neural network for

classification. The high-ranked ones are recommended for delivering reliable and trustworthy communication. The comparative analysis performed using 30,000 test samples illustrated the outperformance of the proposed work in terms of precision, recall, f-measure and accuracy analysis. The simulation analysis demonstrates that the average improvement of the proposed work remained between 9% and 12% which proves the success of the proposed work. Thus, achieving the main goal of research by creating a novel algorithm architecture for trust generation in the Social Cloud Network (SCN). Although complexity analysis plays a significant role in algorithmic research, the main focus of our work is on designing and implementing a workable solution to deal with the crucial problem of service provider selection. Our focus on algorithmic enhancement and trust generating techniques is intended to establish a solid groundwork for future research that delves further into the computational difficulties and performance measures. Using this strategy, we are able to provide a useful and relevant resolution to the pressing problem in social cloud computing.

Author contributions

Conceptualization, SK and SKG; methodology, SK; software, SK; validation, SK and SKG; formal analysis, SK; writing—original draft preparation, supervision, SKG; funding acquisition, SK. All authors have read and agreed to the published version of the manuscript.

Data availability

The data will be made available on request to the corresponding author.

Conflict of interest

The authors declare no conflict of interest.

References

1. Shirvani MH, Rahmani AM, Sahafi A. A survey study on virtual machine migration and server consolidation techniques in DVFS-enabled cloud datacenter: Taxonomy and challenges. Journal of King Saud University—Computer and Information Sciences. 32(3): 267-286. doi: 10.1016/J.JKSUCI.2018.07.001
2. Priyadarshinee P, Raut RD, Jha MK, et al. Understanding and predicting the determinants of cloud computing adoption: A two staged hybrid SEM - Neural networks approach. Computers in Human Behavior. 2017; 76: 341-362. doi: 10.1016/j.chb.2017.07.027
3. Mao C, Lin R, Xu C, et al. Towards a Trust Prediction Framework for Cloud Services Based on PSO-Driven Neural Network. IEEE Access. 2017; 5: 2187-2199. doi: 10.1109/access.2017.2654378
4. Marudhadevi D, Dhatchayani VN, Sriram VSS. A Trust Evaluation Model for Cloud Computing Using Service Level Agreement. The Computer Journal. 2014; 58(10): 2225-2232. doi: 10.1093/comjnl/bxu129
5. Pal K, Karakostas B. A Multi Agent-based Service Framework for Supply Chain Management. Procedia Computer Science. 2014; 32: 53-60. doi: 10.1016/j.procs.2014.05.397
6. Singh S, Chana I. Q-aware: Quality of service based cloud resource provisioning. Computers & Electrical Engineering. 2015; 47: 138-160. doi: 10.1016/j.compeleceng.2015.02.003
7. Ghosh N, Ghosh SK, Das SK. SelCSP: A Framework to Facilitate Selection of Cloud Service Providers. IEEE Transactions on Cloud Computing. 2015; 3(1): 66-79. doi: 10.1109/tcc.2014.2328578
8. Bothra SK, Singhal S. Nature-inspired metaheuristic scheduling algorithms in cloud: a systematic review. Scientific and Technical Journal of Information Technologies, Mechanics and Optics. 2021; 21(4): 463-472. doi: 10.17586/2226-1494-2021-21-4-463-472
9. Singh H, Tyagi S, Kumar P, et al. Metaheuristics for scheduling of heterogeneous tasks in cloud computing environments: Analysis, performance evaluation, and future directions. Simulation Modelling Practice and Theory. 2021; 111: 102353. doi: 10.1016/j.simpat.2021.102353
10. Dalal S, Nagpal S, Dahiya N. Comparison of Task Scheduling in Cloud Computing Using various Optimization Algorithms. Journal of Computer and Information Systems. 14(4): 43-57.
11. Caton S, Dukat C, Grenz T, et al. Foundations of Trust: Contextualising Trust in Social Clouds. 2012 Second International Conference on Cloud and Green Computing. doi: 10.1109/cgc.2012.89
12. Macías M, Guitart J. Analysis of a trust model for SLA negotiation and enforcement in cloud markets. Future Generation Computer Systems. 2016; 55: 460-472. doi: 10.1016/j.future.2015.03.011

13. Yan Z, Li X, Wang M, et al. Flexible Data Access Control Based on Trust and Reputation in Cloud Computing. IEEE Transactions on Cloud Computing. 2017; 5(3): 485-498. doi: 10.1109/tcc.2015.2469662
14. Wang H, Yang D, Yu Q, et al. Integrating modified cuckoo algorithm and creditability evaluation for QoS-aware service composition. Knowledge-Based Systems. 2018; 140: 64-81. doi: 10.1016/j.knosys.2017.10.027
15. Zanbouri K, Jafari Navimipour N. A cloud service composition method using a trust‐based clustering algorithm and honeybee mating optimization algorithm. International Journal of Communication Systems. 2019; 33(5). doi: 10.1002/dac.4259
16. Lee LS, Brink WD. Trust in Cloud-Based Services: A Framework for Consumer Adoption of Software as a Service. Journal of Information Systems. 2019; 34(2): 65-85. doi: 10.2308/isys-52626
17. Kumar R, Tripathi R. DBTP2SF: A deep blockchain‐based trustworthy privacy‐preserving secured framework in industrial internet of things systems. Transactions on Emerging Telecommunications Technologies. 2021; 32(4). doi: 10.1002/ett.4222
18. Dhillon P, Singh M. An ontology oriented service framework for social IoT. Computers & Security. 2022; 122: 102895. doi: 10.1016/j.cose.2022.102895
19. Kumar S, Goyal SK. Swarm Intelligence Based Data Selection Mechanism for Reputation Generation in Social Cloud. In: Proceedings of the 2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON). doi: 10.1109/com-it-con54601.2022.9850947
20. Bangui H, Buhnova B, Ge M. Social Internet of Things: Ethical AI Principles in Trust Management. Procedia Computer Science. 2023; 220: 553-560. doi: 10.1016/j.procs.2023.03.070
21. Ouechtati H, Nadia BA, Lamjed BS. A fuzzy logic-based model for filtering dishonest recommendations in the Social Internet of Things. Journal of Ambient Intelligence and Humanized Computing. 2021; 14(5): 6181-6200. doi: 10.1007/s12652-021-03127-7
22. Mohana SD, Prakash SS, Krinkin K. CCNSim: An artificial intelligence enabled classification, clustering and navigation simulator for Social Internet of Things. Engineering Applications of Artificial Intelligence. 119(2023): 105745. doi: 10.2139/ssrn.4642197
23. Social Internet of Things. Available online: http://www.social-iot.org/ (accessed on 18 October 2022).
24. Jain AK. Data clustering: 50 years beyond K-means. Pattern Recognition Letters. 2010; 31(8): 651-666. doi: 10.1016/j.patrec.2009.09.011