Hindawi
Journal of Computer Networks and Communications
Volume 2018, Article ID 4505290, 17 pages
https://doi.org/10.1155/2018/4505290

Research Article

Approaches to Addressing Service Selection Ties in Ad Hoc Mobile Cloud Computing

Ayotuyi Tosin Akinola and Matthew Olusegun Adigun

Department of Computer Science, Centre of Excellence at University of Zululand, Private Bag X1001, Ongoye, KwaDlangezwa 3886, South Africa

Correspondence should be addressed to Ayotuyi Tosin Akinola; ruthertosin@gmail.com

Received 1 September 2017; Revised 6 November 2017; Accepted 9 November 2017; Published 15 February 2018

Academic Editor: Youyun Xu

The ad hoc mobile cloud (AMC) allows mobile devices to connect together through a wireless connection or any other means and send a request for web services from one to another within the mobile cloud. However, one of the major challenges in the AMC is the occurrence of dissatisfaction experienced by the users. This is because there are many services with similar functionalities but varying nonfunctional properties. Moreover, another resultant cause of user dissatisfaction being coupled with runtime redundancy is the attainment of similar quality computations during service selection, often referred to as "service selection ties." In an attempt to address this challenge, service selection mechanisms for the AMC were developed in this work. This includes the use of selected quality of service properties coupled with user feedback data to determine the most suitable service. These mechanisms were evaluated using the experimental method. The evaluation of the mechanisms mainly focused on the metrics that evaluate the satisfaction of users' interest via the quantitative evaluation. The experiments affirmed that the use of the shortest distance can help to break selection ties between potential servicing nodes. Also, a continuous use of updated and unlimited range of users' assessments enhances an optimal service selection.

1. Introduction

Service-oriented computing (SOC) is an emanating interdisciplinary paradigm for realizing a distributed computation, thereby changing the traditional pattern of incurring the cost of access to unwanted software applications, thus depicting a better service delivery on various service request provisioning schemes [1]. The software applications that are made available in the SOC are primarily a product of fundamental resources which are *services*. The services are self-describing and computational resources that enhance an automatic, rapid, and low-cost provisioning of software applications in a distributed environment. Due to the explosive growth of various mobile applications, the Internet serves as a linking medium between the mobile clients and the service providers. Therefore, with the view to achieving agility and flexibility between provider and client interaction, the SOC has been generally adopted. The SOC assumed business functions as modular packages in the form of services that run on any service-oriented architecture (SOA) [2].

Moreover, the cloud computing paradigm is an example of the SOA that serves as an information technology servicing model where computing services are delivered on demand to customers over a network in a self-service fashion, independent of devices and locations [3]. Thus, the resources required to provide the requisite quality of service levels are shared, dynamically scalable, rapidly provisioned, and virtualized with minimal service provider interaction. Users pay for the service as an operating expense without incurring any significant initial capital expenditure, with the cloud services employing a metering system that divides the computing resources into appropriate blocks.

There are three basic processes that are involved in the provisioning of services in any service-oriented platforms which include service discovery, service selection, and service composition [2, 4, 5]. By the discovery of services, we refer to the process of identifying potentially available web services which can carry out a particular task. However, among the available web services, there would certainly be a particular service in the midst of the discovered web services which best satisfies the request of the present service user at the time of request based on the quality of service that was specified. This process is termed "web service selection." The composition of web services involves the integration of two or more web services together to implement interconnected tasks. For example, this interconnected task can be a web service which searches for a flight schedule as well as another that locates a meter taxi or a hotel reservation. This kind of request is being carried out via a composite web service from the provisioning platform.

These three concepts are very important in any service provisioning platform and thus become one of the central focus to research scholars. Therefore, this calls for adequate mechanisms to be deployed to enable these processes to run smoothly in the course of providing optimal services on the aforementioned platforms. Addressing these three concepts poses various challenges in the modern service provisioning platforms today especially the concern with the enormous release of web services that are performing similar functionalities. In the same vein, when mobile devices were connected to the cloud platforms for service usage, it becomes imperative that optimal service selection is addressed to avoid the dissatisfaction with service responses to the intended users. A typical example of such a mobile service consumption scenario is the mobile cloud computing paradigm. This paradigm enables the use of mobile devices to consume resident web services within the cloud computing architecture [6–9]. The dissatisfaction becomes very tasking in a situation where the users eventually become the platform for service provisioning in the case of the AMC system, wherein mobile users are constantly in a move while consuming the web service [10].

Several parastatals such as military and other forms of organizations which are involved in emergency operations have been adopting the AMC technology towards addressing the pressing need for total avoidance of interconnectivity issues, both in less-developed areas as a result of bad infrastructure and even in developed places due to natural disasters, etc. [11]. For example, in an emergency operation which utilizes an AMC setup, the nearest mobile nodes with the required service must be the chosen node for the operation so that the task at hand will be carried out successfully and promptly within the limited allotted time frame. Moreover, the occurrence of service selection ties during the course of selection especially in a range of different service locations within the mobile devices needs to be addressed as this has a lot to speak to the measure of satisfaction that the users experience within such a system [12, 13].

A contemporary project that is currently under development, which looks into most of the aforementioned issues, is GUIISET (GRID-based utility infrastructure-infrastructure less for SMME-enabled technology) from the Centre of Excellence of the University of Zululand in South Africa. It is an advancement on the previous GUISET project through the incorporation of this infrastructure less (ad hoc model) platform into the system [14–17]. The goal of this incorporation is to enable the developed platform to be able to render services to mobile devices especially in the context of m-Health, m-Learning, and m-Commerce.

Several works have been done which ranges from GUISET's implementation [18–20] to its performance evaluation [17, 21, 22] and security [15, 16, 23] as well as the Pricing strategy to be used on the platform [14]. However, little research has been conducted in relation to providing an optimally satisfactory service especially in the newly integrated ad hoc model system. In addition, issues relating to similar computation scores (selection ties) which often inform unsatisfactory service usage have not been fully addressed in most of the articles to the best of our knowledge. Hence, this article contributes to the existing knowledge via providing solutions to "service selection ties" in both mobile nodes and web service provisioning to various patronizing customers via deployable selection mechanisms. Moreover, service response delay is also addressed to reduce the time taken to deliver a selected service to intended users.

The remainder of this article is as organized as follows. Section 2 reviews various service selection approaches in different platforms to bring out the gap that is missing which is very important on the AMC. The section also reviews the earlier attempts to address the issue of service selection in various platforms. Section 3 discusses the QoS property resolution strategy, while Section 4 enumerates the modes of selection occurrence in the AMC. Section 5 addresses the selection ties in AMC mobile nodes using the multidynamic distance-based approach, where a typical surgical emergency service is used as a scenario. Section 6 discusses the feedback-based selection approach for optimal web service selection in an AMC, while Section 7 explains how the performance evaluation was conducted with some discussions. Section 8 explains the conclusion and future work.

2. Literature Review

This section elucidates the methodologies of the service selection process with a view to discussing various methods that can be deployed for selection of services in general from any service-oriented computing platform. Moreover, the selection approaches that are often deployed in an ad hoc environment were also discussed. Hence, we shall be explaining service selection methodologies and service selection approaches in the AMC. In addition, this section also attempts to review other approaches to optimal selection in the AMC.

2.1. Service Selection Methodologies. The selection methodologies are the various methods that are adopted for selection services within a service-oriented architecture platform. The methodologies used in service selection are generally classified into three according to the work of Swarnamugi [24].

These are explained below:

(1) Functional-based method: This method selects the appropriate service based on the retrieval of a functional description of a service from the registries and then certifies that the description and requirement of the interfaces match each other. There is always a need to convert the web service into the semantic web to enhance the easy description of web service functionality. The semantic web service selection was implemented in the work of Klusch and Kapahnke using the hybrid version of the SAWSDL matchmaker called SAWSDL-MX [2, 25]. The drawbacks of the functional service selection approach make the service provider seek for a more informative alternative that could differentiate their products from others in relation to performance. Moreover, considering the AMC environment, the use of the highly logical algorithm and complex ontological data storage generate a challenge for memory-restrained mobile devices; therefore, the approach is considered unsuitable for deployment.

(2) Nonfunctional-based method: It is a common experience in the service provisioning environment for services to provide similar functionalities with different nonfunctional properties. Hence, such services can only be differentiated by considering their nonfunctional attributes which can either be quality of service or context-based [24]. In [26], the author proposed the architecture for a web service selection using the QoS-based approach in a service provisioning environment where the service user searches the service registry (UDDI) for the list of all services that address the concerned request. The service broker here assists in differentiating the various services in the registry. Thus, the service registry becomes a complex task for an AMC-based environment. Xin and others in [27] considered a framework that combines QoS attributes with user preferences of a group of consumers to propose an algorithm and a mobile service selection model to solve the selection challenge. But the selection was based on a group of collective users which is not individually tailored as expected in an AMC environment. The work of Amoretti and others in [28] proposed a reputation-based selection framework which emphasizes on an intra- and inter-SOP (service-oriented peers) module interaction. The framework contains a component called SAFE that ensures that a mobile peer computes the reputation of a provider based on the previous experience. The SAFE component is assigned the task of "voting strategies" to ensure that proper record of the aggregated reputation influences selection decisions. However, this work was silent about the situation where web service attains similar computation aggregates (otherwise called selection ties) thereby leaving behind gap to fill. Furthermore, the study conducted by Akingbesote and others [18] proposed a quality of service aware Multilevel Ranking Model (MLRANK) for selecting an optimal web service in cloud computing. The study addressed the occurrence of ties within a number of services that are available in the UDDI. The study highlights the challenge of selecting an optimal web service when there are ties with the used criterion where performance alternatives have the same score. The study achieved optimal selection by comparing the service consumer's QoS preference with the web service QoS offerings. The provider offering that best fits the QoS preference is taken as the optimal web service. The study used nondeterministic QoS metrics and concentrated on various information services to test the performance of the proposed model. However, this study serially considers each of the selected qualities one after the other and checks which one is higher than the other to make a selection. This approach is not reliable in the context of the AMC. This is because decisions are expected to be as fast as possible. In addition, the work never considered the possibility of the consumers having relatively equal priority for the specified QoS properties which makes the selection yardstick less efficient. Keidl and Kemper in [29] proposed a context processing framework to influence context-aware web service provisioning. However, the framework uses an insufficient number of context parameters among which are location and client addresses. The location specified by the SOAP message body of the web service is fixed; thus, it is not suitable for a dynamic environment like the AMC system. Since the mobile device keeps changing location, the use of fixed location as one of the parameters will be irrelevant to service selection. Keskes proposed the combination of context and QoS for service selection and considered the effectiveness and consistency of the method in relation to using an increasing number of service parameters more than six [30]. But, the use of the ontological context to decipher the information brings about the issue of ambiguity if deployed to memory-constrained mobile devices for service selection.

(2) User-based selection method: The measure of the trustworthiness of a particular web service is termed "the Reputation." This measure mainly depends on the end users' experience of using the particular service. Various end users may have dissimilar views about the same web service. However, the reputation is expressed as an average ranking that is given to a web service by the end users, thus deriving a range of ranking from these end users. This system determines the QoS of a service provider through calculation of the difference between the published service provider value and the user feedback response. A higher difference value typically shows a lower QoS rating for that particular service provider or web service. The service users are allowed to provide a feedback

through the use of a pair of keys [31]. The users authenticate with the aid of these keys, and they are allowed to update the QoS criteria based on their experience. However, the provider would not always be available to affirm and see to the proper updating of the QoS properties. The high possibility is that both the user and consumer would likely not be available at a similar time, therefore resulting into overloading of the system through unconsumed user requests.

2.2. AMC Service Selection Approaches. The approaches to service selection in the AMC require more criteria to be taken care of because of its inherent nature. This is because in an AMC, there might be the need for the selection of the nearest mobile node to provide a particular service, and at some other times, it could be the web service that is resident within the mobile node that needs to be shared among the AMC mobile devices. Considering the influx of mobile devices within an AMC, there could be a sudden increase in the number of connecting mobile devices which invariably increases the number of service requests; thus, we itemize the following concern with respect to AMC selection approaches: (1) any approach that must be used should cater for sudden increment in service requests [32]. (2) The approach must address the dynamic nature of the mobile environment, and (3). the approach should eradicate the arbitrary selection of mobile nodes and web services during the occurrence of service selection ties which is a major challenge in service provisioning in the AMC environment [18, 33].

According to [34, 35], service selection in the ad hoc mobile environment is carried out using the following approaches: (1) hop-based, (2) QoS-based, and (3) integrated/hybrid-based. The hop count has been used for time-constrained mobile service selection in wireless mesh networks and sensor networks, especially where the response time is very pivotal such as in critical and life-saving situations [36, 37]. This system considers time as a very important issue and therefore looks or searches for the closest node to the requestor. This has the advantages of (i) a lower risk of downlink with the least distances apart; (ii) a lower rate of battery consumption with the reduced distances between the nodes, and (iii) lesser possibilities for the two nodes, to be out of reach of each other [38].

The integrated approach combines the properties from the route/hop count with the QoS approach to arrive at the optimal solution for a particular request. Due to the nature of the request that a service user specified, there might be a need to prefer particular properties more than the other. For example, time constraint might be very imperative for a user in need of urgent medical attention or user who needed to catch up quickly on a flight, whereas for another user that is requesting for an educationally related service might not be as more urgent. Mobile devices enable the use of various services on transit through a wireless connection either to the cloud server or within themselves. Thus, the integrated (hop count, QoS properties, and feedback response) selection approach seeks to address users' satisfaction by ensuring that the optimal service is being selected at every request.

3. QoS Property Resolution Strategy

The impact of quality of service cannot be overemphasized in the process of service selection. It serves as a building foundation upon which other service approaches can be integrated to enhance optimal service selection in any service-oriented provisioning. The selection of service is built by mapping the requests onto a web service among the available ones. The mapping process helps in locating the actual web service. The aggregation of the QoS for service selection is based on the individual preferences with regard to QoS properties. The following two steps assist in ensuring that this process is properly carried out.

3.1. The Scaling Process. This study uses a simple scaling or normalization technique. This concept of scaling ensures even distribution of QoS properties since these properties differ in value from each other with hardly any comparability among them. The range of values derived fully expresses the order and level of competitiveness among the available web services [39]. $\{Ss\} =$ set of ad hoc mobile services of $\{s_1, s_2, s_3, \ldots, s_j\}$ with a set of qualities $\{Q_q\}$ being $\{q_1, q_2, q_3, \ldots, q_m\}$. This implies that when a sample of the AMC web service is selected such as s_i, it will have a corresponding set of qualities such as $\{q_{i1}, q_{i2}, q_{i3}, \ldots, q_{ij}\}$. Now, assume that the value of the minimum and maximum ith QoS properties within all the available services are $Q_j^{\min}$ and $Q_j^{\max}$. For example, the general quality of some QoS properties decreases with an increase in the value of the property such as service cost and response time; thus, by scaling, we try to minimize the property as much as possible using the normalization equation (1) according to [40] as

$$\acute{V}_j = \begin{cases} \dfrac{Q_j^{\max} - Q_{i,j}}{Q_j^{\max} - Q_j^{\min}} & \text{if } Q_j^{\max} - Q_j^{\min} \neq 0 \\ 1 \quad \text{if } Q_j^{\max} - Q_j^{\min} = 0. \end{cases} \tag{1}$$

In a similar manner, other services whose QoS properties increase with an increasing value of such a property such as availability and reliability are all normalized using the expression in (2) according to [38] as

$$\acute{V}_j = \begin{cases} \dfrac{Q_{i,j} - Q_j^{\min}}{Q_j^{\max} - Q_j^{\min}} & \text{if } Q_j^{\max} - Q_j^{\min} \neq 0 \\ 1 \quad \text{if } Q_j^{\max} - Q_j^{\min} = 0. \end{cases} \tag{2}$$

An interval of the range of [0, 1] is produced from (1) and (2), where "1" represents the best QoS property, while "0" represents the least QoS property, thereby reflecting any form of changes that occur to the general quality of any of the services whether it decreases or increases. The algorithm for normalization process is as shown in Algorithm 1.

From Algorithm 1, a new matrix V was generated:

$$V = \left(V_{i,j}; 1 \leq i \leq n, 1 \leq j \leq m\right). \tag{3}$$

```
Input:
   A set of qualities of a Web Service $\mathbf{S(t)} = \{s_1, s_2, \ldots s_m\}$
   that each describes a service
Output:
   A matrix of normalized QoS parameters
Step 0: Initialization, create m by 5 matrix P
Thus, having:

for $(i = 0; i < m; i + +)$do
for $(j = 0; j < n; J + +)$do
if (qf $[j]eq0$)
if (diffqos $[j]! = 0$)
$v[i][j] \leftarrow ((\text{qmax}[j] - p[i][j])/\text{diffqos}[j]);$

else
   $v[i][j]$
endif
else if (qf [j]eq1)

if (diffqos [j]! = 0)
$v[\text{i}][\text{j}] \leftarrow ((p[i][j] - \text{qmin}[j])/\text{diffqos}[\text{j}]);$


else
   $v[i][j] \leftarrow 1$
endif
return
   $V = (V_{i,j}; 1 \leq i \leq n, 1 \leq j \leq m;)$
endif
   endif
```

Algorithm 1: QoS property normalization algorithm.

3.2. Weighted Summation Assignment. There is a need to assign weight to the respective normalized value of each of the service users so that the preferences and the importance of a particular quality can be expressed by the user when requesting a service. Therefore, treating the pool of services as a sample space with coordinates that are having origin as a vector, the services can be represented as

$$\vec{S} = \acute{V}_1\widehat{V}_1 + \acute{V}_2\widehat{V}_2 + \acute{V}_3\widehat{V}_3 + \cdots + \acute{V}_m\widehat{V}_m, \tag{4}$$

where the service is represented by $\vec{S}$ and $\acute{V}_1$, $\widehat{V}_j$; both represented normalized QoS properties as well as the unit vector along the axis of the normalized property. Supposing the mean quality of service is given by S_{mean} and if $\acute{V}_{\text{mean}}$ represents the mean value of the *i*th QoS property, then,

$$\begin{aligned}\vec{S}_{\text{mean}} = {} & \acute{V}_{1\,\text{mean}}\widehat{V}_1 + \acute{V}_{2\,\text{mean}}\widehat{V}_2 \\ & + \acute{V}_{3\,\text{mean}}\widehat{V}_3 + \cdots + \acute{V}_{m\,\text{mean}}\widehat{V}_m.\end{aligned} \tag{5}$$

Considering the mean value of the normalized qualities, we have the summation of the means ranging towards 0.5 as $1 \geq \acute{V}_j \geq 0$; hence, we can rewrite the expression to be

$$\widehat{S}_{\text{mean}} = \frac{1}{\sqrt{m}}\left(\widehat{V}_1 + \widehat{V}_2 + \widehat{V}_3 + \cdots + \widehat{V}_j\right). \tag{6}$$

Carrying out the dot product of the *i*th service of $\vec{S_i}$ as it projects over the mean $\vec{S}_{\text{mean}}$, then we have the expression

$$\vec{S_i} \cdot \vec{S}_{\text{mean}} = \frac{1}{\sqrt{m}}\left(\acute{V}_1 + \acute{V}_2 + \acute{V}_3 + \cdots + \acute{V}_j\right). \tag{7}$$

Hence, the summation of quality of service properties can be expressed as

$$\text{Summation of quality properties} = \frac{1}{\sqrt{m}}\sum_{i=1}^{m}\acute{V}_j. \tag{8}$$

Assuming the user request for a special preference in any of the qualities of service that are highlighted, the summation of the preference in terms of weight assignments could be expressed for the *i*th QoS property; thus,

$$\text{Weighted summation assignment} = \frac{1}{\sqrt{m}}\sum_{i=1}^{m}\acute{V}_j * W_i. \tag{9}$$

The output generated from this normalization and weighted summation assignment often results in several web services and service nodes with similar aggregate scores [18]. This challenge shows that there is a need for a mechanism that helps to inform optimal service selection through breaking the occurrence of selection ties in the heterogeneous web service provisioning.

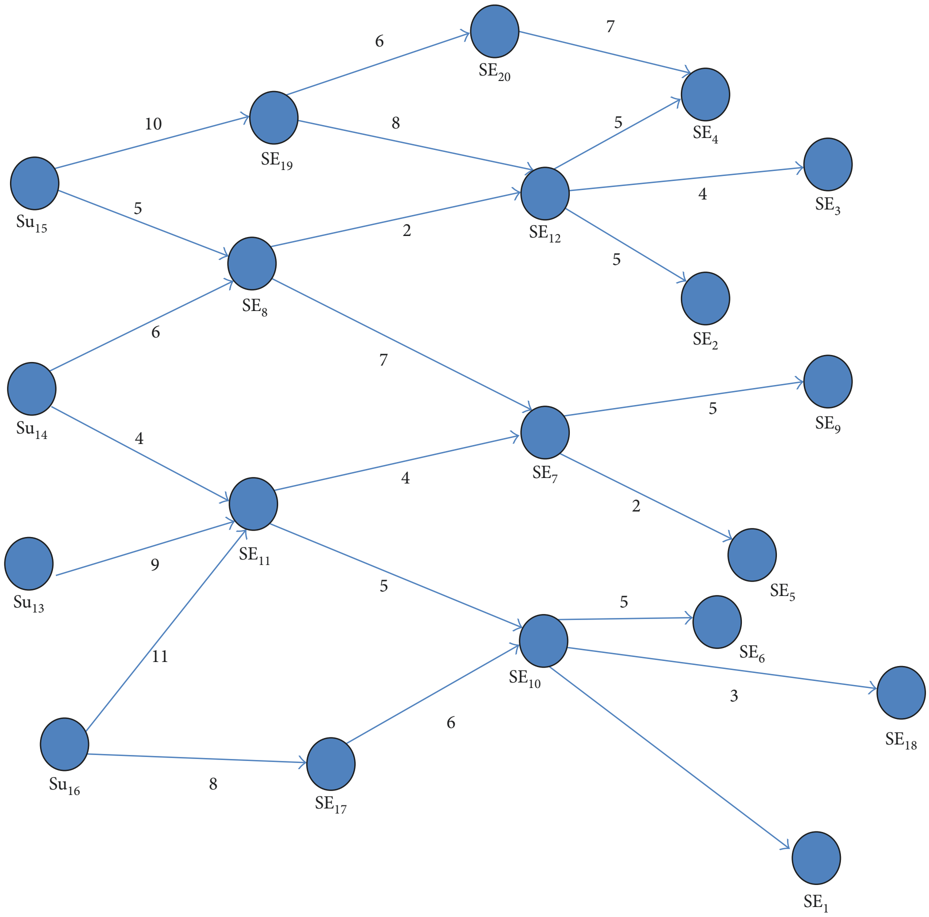

Figure 1: m-Health scenario routing traffic from the source to the sink.

4. Modes of Selection Ties in the AMC

Selection ties occur in a situation where there are two or more "*objects*" that qualify to attend to a particular request from a service consumer [11, 18]. In the case of the AMC, a mobile node can be a service provider where it helps to forward a traffic via the shortest route mechanism to the traffic destination. In addition, the resident web services within the mobile nodes in an AMC can attain similar computation scores via the processes discussed in Section 3. The literature currently selects any of the mobile nodes or web services that attain similar computation scores to address the service request which in turn is usually not the optimal option. This invariably results in some level of dissatisfaction on the path of service consumers when at other times, a close friend was able to experience a better service delivery than them due to random selection at "ties" occurrence.

For example, let us consider a case where teachers and students formed an AMC within the school premises, in which they all consume similar educational resources. The experiences of the teachers and the students will virtually be different from one another in situations where there are similar computation scores in the selection process. Therefore, to avoid this kind of situation, this article addresses two major modes of "*selection ties*" occurrence in the AMC:

(A) The mobile node selection ties

(B) The web service selection ties

This article offers substantiated mechanisms with respect to these two modes towards the selection of an optimal service provider to various service requests that are issued in the context of the AMC environment.

5. Multidynamic Distance-Based Approach

The term *multidynamic* distance in this context refers to the use of QoS properties alongside with distances apart among the mobile nodes, otherwise referred to as the shortest route in the course of service provisioning. The approach here is deplorable in an emergency state especially in a typical m-Health situation as shown in Figure 1. The SU$_{\text{nodes}}$ is the principal service users.

The traffics are expected to take the shortest route to the destination while bearing in mind that the intended nodes have the capacity to offer the required services. The hypothetical quality of service parameters is as shown in Table 1, containing the prices, response time, reliability, and availability of the mobile nodes. The last column in the table also depicts the summation of route distances apart for the intended traffic. Suppose that a service user (SU$_{15}$) in Figure 1 is issuing a request for a surgical emergency service (SE$_x$)

Table 1: Quality of service properties.

Services			Properties		
SE services	Price (R)	Response time (ms)	Reliability (%)	Availability (%)	Route (m)
SE_1	82	200	69	75	29
SE_2	70	360.0	20.2	43	12
SE_3	75	330.0	36.5	50	11
SE_4	74	310.10	31.6	41	12
SE_5	81	236.5	35.7	48	14
SE_6	80	210	40.3	60	25
SE_7	82	208	67	72	12
SE_9	82	210	48	78	17
SE_{10}	73	250	35	45	20
SE_{12}	67	233	60	88	7
SE_{17}	55	214.5	55	48	26
SE_{18}	78	305	67	53	23
SE_{19}	76	300	82	64	10
SE_{20}	80	295	45	81	16

```
Set parameters ($p$, $r_t$, $r$)
Input consumers required QoS
    Call normalization
    Set weights
Calculate score
    if Ss(Score) > 1 then
call DP
Print Ss(Sortestroot)
    else
print Ss(score)
    end if
end.
```

Algorithm 2: Multidynamic algorithm.

from any mobile nodes (all SE_x) who can render such a service. We implemented the multidynamic algorithm that was depicted in Algorithm 2. We used the normalization algorithm to analyze the considered qualities of service (price, response time, reliability, and availability), which generated the normalization results that were depicted in Table 2. These outputs are the results that are fed into the weight assignment stage to determine the interest of the service consumer who requests for the service of the mobile nodes.

Algorithm 2 calls for the computation of the normalization and weight assignment and the calculation of aggregate scores. The occurrence of ties further initiates the call to Algorithm 3 which invariably differentiates the services with similar scores. The "*Sortestroot*" uses the dynamic programming algorithm to effect the choice of the shortest route that needs to be calculated in the process. It considers only the mobile nodes with the same computation or aggregate scores as shown in Table 3 and Figure 2. The algorithm selects the mobile nodes among those with similar computation and further selects the one closest to the requesting node for user consumption.

```
$v$ = vertex and $n$ = node
Input ($v$, $n$)
Shortestroot ← 0
Array Shortpernode ← 0
Shortestroot ← 0
Shortrootest ← 0
Repeat
  For $i = v - 1$ to $v$
    For $j = 1$ to $n$
      For $k = 1$ to edgeno
    Shortpernode[$k$] = search min cost {$u$, $v$}
      Next $k$
      Sort Shortpernode[$1 - k$] in descending
        if
              shortrootest > Shortpernode [1]
        then
        shortrootest = Shortpernode [1]
      endif
    Next $j$
    Shortestroot ← Shortestroot + shortrootest
    $v = v - 1$
  Next $i$
Until $v = 1$
```

Algorithm 3: Shortest route DP algorithm.

5.1. Selection of Optimal SE Service. The selection of the surgical emergency service (SE) that meets the requested QoS is selected by first providing ranges of weights that depict the amount of priority given to each QoS which ranges from 0 to 1. This weight represents the degree of importance associated with a specific QoS property, and they are fractions whose sum must be equal to 1. For example, if a consumer's topmost priority is on price for surgical emergency service, then the higher weight is given to price. The other properties take lesser fractions, respectively, till the least of them. The service that provides the best utility level based on

Table 2: Normalization of SE services.

Services	Properties				
SE services	Price (R)	Response time (ms)	Reliability (%)	Availability (%)	Route (m)
SE_1	0.8181	0.2373	0.6175	0.9290	29
SE_2	0.2727	0.064	0.0802	0.7828	12
SE_3	0.5	0.2523	0.3843	0.6829	11
SE_4	0.4545	0.3763	0.2929	0.8942	12
SE_5	0.7727	1.1064	0.9533	0.5620	14
SE_6	0.7272	1	0.4552	0.7296	25
SE_7	0.8181	0.9688	0.3694	0.3827	12
SE_9	0.8181	1	0.8504	0.9284	17
SE_{10}	0.4328	0.4270	0.3894	0.530	20
SE_{12}	0.45	0.8639	0.8693	0.7829	7
SE_{17}	0.8012	0.4390	0.5728	0.7328	26
SE_{18}	0.6013	1	0.3874	0.4038	23
SE_{19}	0.520	0.2089	0.8429	0.7829	10
SE_{20}	0.3720	0.7823	0.7832	0.5390	16

price becomes the chosen one provided that it is the only one. The weighted summation assignment generates the aggregate score as given in (9).

When the number of ad hoc surgical emergency services that have the maximum score is greater than one, then we consider the route to each of the destination. We then applied the dynamic programming as shown in Algorithm 3. This is done by describing our data elements needed in the form of cities and distances. We then formulate this as an optimization problem, and the minimum route is determined based on the minimum cost tour.

The idea in this section has been used in our previous paper [11], but we considered a larger number of mobile nodes in this experiment; thus, this solution becomes one of the solution approaches under the present article. The detailed comparison among the mobile nodes with aggregate scores was depicted in Figure 2. Other approaches with respect to this particular scenario select any of the mobile nodes among those that attain the request of the service user, but a better utility level is guaranteed with this mechanism in place for mobile service selection.

Table 3: Shortest route-based selection.

Services	Properties	
SE services	Route (m)	Aggregate scores
SE_1	29	0.7830
SE_2	12	0.7828
SE_{12}	**7**	**0.7827**
SE_{20}	16	0.7829

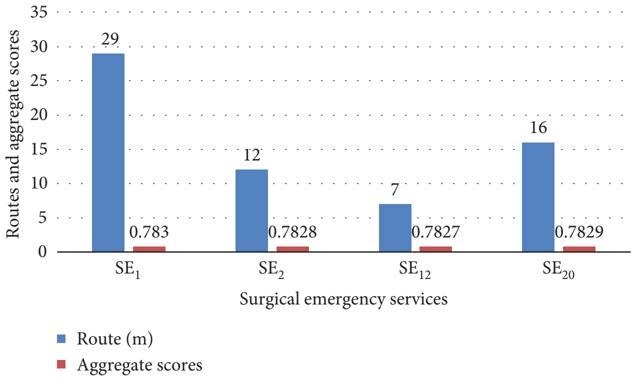

Figure 2: Shortest route service selection in critical situations.

6. Feedback-Based Approach for Web Services

The user feedback has been an approach that is generally used for improving the performance of a system. This is due to the fact that it gives an updated and continuous information about the performance of an application (web services) with a view to making informed decisions about the behaviour of such an application. In this section, we deployed the use of user feedbacks to enhance better service selection that eradicates the occurrence of web service ties during dynamic service selection. Again, citing the case of teachers and students who team up to form a group as an AMC that exists within a school environment where both participants share educational materials, the challenge that is often experienced is that we want the system to be able to render the optimal web service that guarantees the highest level of satisfaction each time to the service user. Thus, we propose a middleware engine that collects the users' view after consuming web services and use it to range the performance of the resident web services on the system to enhance an optimal selection in future requests. The diagram in Figure 3 shows the interrelated components for effective service selection within an AMC.

The main purpose of introducing the feedback mechanism is to eradicate the possibility of service selection ties within the AMC. The setup mechanism majorly contains the rater, the feedback composer, and the database wherein the

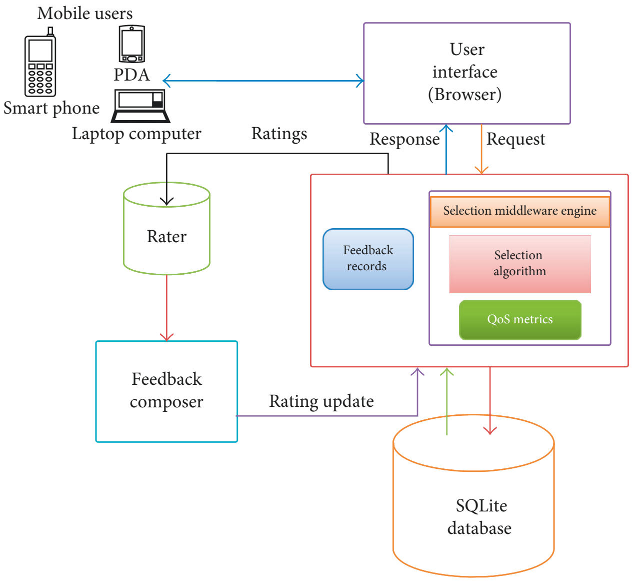

Figure 3: Rating mechanism in AMC service selection.

feedback records are kept. Most of the previous steps such as the normalization and the weight assignment are already assumed to have been carried out in this stage. We also assumed that various users provide the system with truthful responses. The rating process follows two major steps which are the collection of the user feedback and the prediction of the appropriate services for the service requestor.

6.1. Collection of User Feedback. The influx of the AMC with various forms of web services ultimately results in the occurrence of similar ties within the environment. When the similar ties occur, the instruction is to select any out of those ones to carry out the task, and in the absence of such an instruction, the system goes into a temporary redundancy state in which indecision sets in. Eradicating the challenge of similar ties in service selection will enhance prompt response to users through overcoming the possibility of occurrence of redundancy states [41]. The user feedback system uses the credibility level system to collect, process, and rank the responses from the users. Thus, five categories of credibility trust on web services were assigned, namely, the most credible (l_5), more credible (l_4), credible (l_3), less credible (l_2), and the least credible (l_1). The associated strength to each user response can be normalized in

$$\sum_{i=1}^{5} l_i = 1. \tag{10}$$

Using this definition, the researcher assigned the value of 0.2 to express the least credibility value l_1 which shows a lack of trust regarding the selected service. Other values starting from 0.4 to 0.6 are assigned to l_2 and l_3, respectively, where they both typify a low credibility trust. The last sets of values of 0.8 and 1.0 which represent l_4 and l_5 express reliable trust from the service users. Every rating selection made by the user is computed by finding the average or mean of the feedback values selected. This is used to update the system for the latest rating regarding a particular service. This credibility trust utilized in the proposed system is coupled with a recommendation technique that will assist to make the right decision for the service user [41]. The recommendation technique that is adopted is an item-based collaborative system which is a subset of the collaborative filtering approach. This recommendation system is important in this context for two major reasons:

(1) It identifies the best of the peer of web services among those attaining similar aggregate scores.

(2) It helps to compute final selection based on the services used by the requesting user, provided the user has invoked services in the system before.

The item-based collaborative approach helps addresses these two highlighted points thereby helping to carry out the prediction creation tasks on web services with the similar aggregate scores.

6.2. Prediction Creation. This section of the service recommendation predicts the best web services based on the previous ratings recorded by the system from earlier service users. The prediction proposed uses the binomial probability density distribution. In this approach, the following variables were used as expressed below:

Let k be the context of service selection such as m-Learning.

X_k is the user's rating for a service in context k.

P_k is a user's preference for a service in context k.

The feedback composer rates each service to select the optimal service for users according to the following density for random variables X_k for which x_k is a typical instance:

$$X_k \sim \frac{1}{n} \text{Binomial}\left(n, P_k\right), \tag{11}$$

where $n =$ no. of mobile devices (nodes).

$$\begin{gathered} E\left(X_k\right) = P_k, \\ \operatorname{Var}\left(X_k\right) = \frac{1}{n} P_k \left(1 - P_k\right). \end{gathered} \tag{12}$$

We deployed the user rating into the probability prediction for selection. By expressing the ranges over the binomial distribution, the summary was explained in two ways:

(1) When the feedback is very low, the mean of the rating distribution X_k should correspond to a very low value. For example, if the credibility from the user based on quality experienced is l_2, then the feedback composer normalizes the probability to assign $E(X_k) = 0.2$.

(2) Contrariwise, when the feedback is high, $E(X_k)$ is also expected to be high. Thus, the feedback composer chooses $E(X_k) = 1$ when the response is high.

This study implements the following variation on the mean X_k; thus,

$$E(X_k) = \mu_k = P_k + 2\left(q_k - \frac{1}{2}\right)(q_k - P_k). \quad (13)$$

Every service user draws a rating from the mean X_k for every service that has been rated. These sets of all users' ratings are the inputs to the composer for proper selection to take place. The result of (13) produces series of mean distributions with each of the provided users' ratings, thus aligning the results into a format of a binomial distribution curve. This is because it continually finds the mean of the ratings submitted by the users which always tends to reach the maximum value of ratings as given by a typical binomial distribution diagram curve. The ratings maintain their value if similar values are provided by the new users who make no difference from the earlier value in the feedback record. But whenever the value of the new ratings is more than the previous record, the rating increases. Thus, the feedback rating falls and rises along the binomial distribution curve. The peak of the dome-shaped top of the binomial curve corresponds to the least rating from the users, and such services are not often predicted to users for use because of the low value or rating which shows that the service performs poorly within the specified service functionality. The flattened edge tending towards 1.0 shows the best services to be predicted to the user for use, and the higher it moves from the dome-shaped curve towards the flattened end, the better the web services that are predicted for use in terms of QoS.

6.3. Environmental Specification. To test this proposed approach, this work deploys the use of the Sun Wireless Toolkit 2.5 Beta Version J2ME from the Sun Microsystems packages to test the behaviour of the selection mechanism. The SQLite database is shown outside of the physical connection setup according to Figure 4 to depict the three-layer network model setup (consumer layer, middle layer, and database layer); however, the DB was actually embedded within the server machine. The emulator interface design is developed to run on the J2ME-enabled devices such that it can be easily deployed on real smartphone mobile devices that support communication through the Hypertext Transfer Protocol interconnections.

The implementation of the server is accomplished by the use of J2EE-compliant servers such as Glassfish Application Server 3.1.2. The server-side components are configured to run on any server that conforms to the J2EE specification. The server handles messages from clients through the use of WMA Bridge API, which enhances message interaction during client service invocation from the server. The server contains the web component which runs within the servlet container and also uses the JAXP to communicate with the external data sources. The external data sources originate as information from the XML database, which is a collection of mobile web services.

The whole setup was tested using a mobile laptop running on Windows 7 Professional Edition as an operating system. The laptop had an Intel Core(TM) *i*7-4500U processor with a processing speed of 2.40 GHz and 8.00 GB RAM. The application consumed 10.4 MB of the hard drive storage which is favourable to the mobile computing environment due to low memory consumption. Figure 4 also contains the mobile emulators developed from the Netbeans IDE together with the Apache JMeter load generator. The emulator shows the typical interface of the nature of service requests which is similar to requests generated by a typical Apache JMeter for the purpose of testing the performance of the proposed selection model. Figure 5 shows a typical emulator interface for service specification and query interface. The Apache JMeter allows the setting of network parameters like latency and jitters, during the experiments. Therefore, this setup mimics the AMC service provisioning environment for testing the performance of the deployed mechanism.

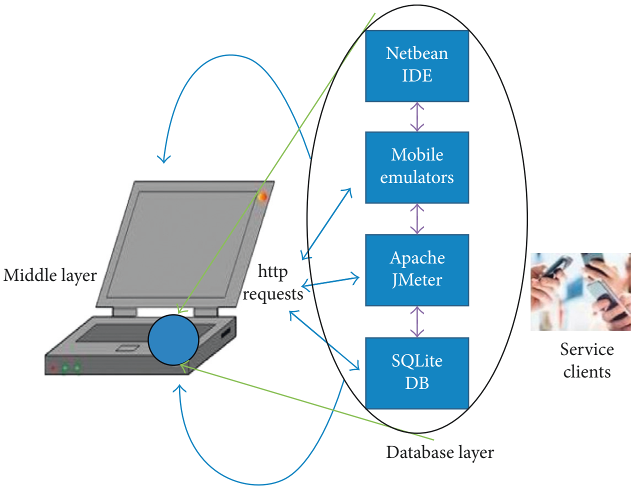

Figure 4: Mechanism simulation setup.

7. Performance Evaluation

The performance of the selection mechanism on service selection ties was tested with a series of experiments to ensure that the concerned challenge is being addressed. This article first evaluated the selection system using certain parameters such as throughput and service availability to check up that the system is operating normally under a good condition. We later conducted experiments to determine the effect of user feedback on the selection process that involves selection ties and as well use a case of a real-life scenario where service selection ties occur to see how ties' breaking of services might occur before offering the service to the users.

7.1. Service Availability Rates and Throughput Performance. There are many options for quantifying the rate of service availability within interconnected systems. One of the popular options is the use of statistical analyses [42–44]. These approaches are more feasible in a very stable environment where mobility is not occurring too often. Thus, the constant dynamic nature of the AMC will not

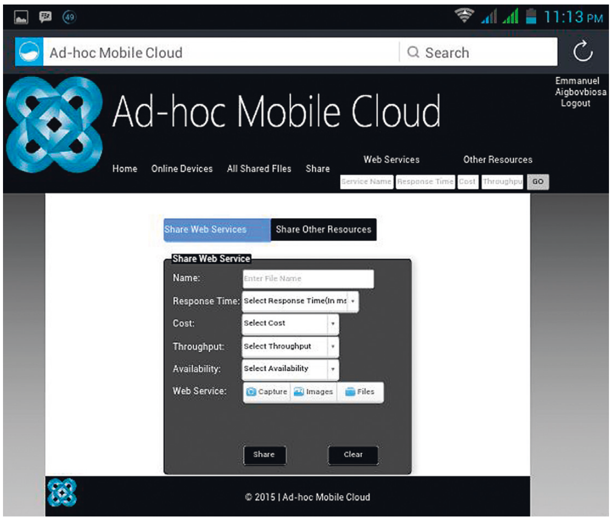

Figure 5: Service specification and query interface.

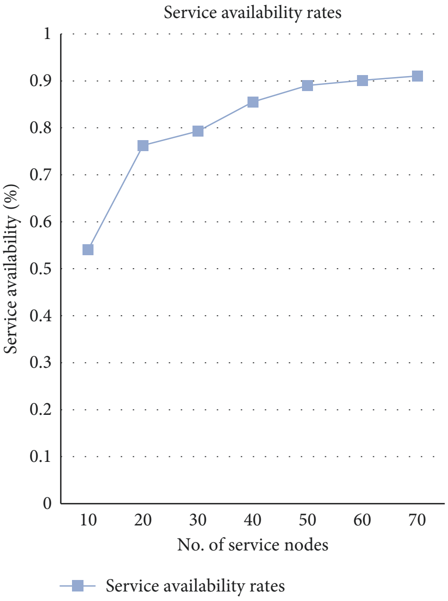

Figure 6: Service selection effect on web service availability.

favour these approaches. However, in this experiment, we assigned some weight indicators to certain parameters (fundamental resource availability, i.e., HTTP authorization manager, response assertion, HTTP response default, and Hypertext Markup Language (HTML) link parser) in the Apache load generator which is indicated at runtime by assigning values ranging from 0 through 1 for each of the parameters. Based on these parameters, suppose P_{xy} represents the weight of set parameters S_{xy}, where S_{xy} sums up the indicator values I_{xy} , and given that

$$\sum_{x=1}^{x=a} I_{xy} = 1 , \tag{14}$$

then the service availability of the AMC is therefore calculated using the expression:

$$\text{Ava} = \sum_{x=1}^{a} \sum_{y=1}^{b} P_{xy} S_{xy}. \tag{15}$$

As the number of mobile nodes in the AMC is increased, there is a relative increase in the number of requests and it invariably has the corresponding impact on the percentage number of service availability rates as shown in Figure 6. From the graph shown, it can be deduced that as the number of service provider nodes increases, the service availability climbed steeply at the initial stage but increased more at 20 nodes (i.e., as the nodes doubled). This can be interpreted to mean that all things being equal, the greater the number of service providers, the more the number of services available for consumption within the system. It was noticed that as the number of nodes increases within the AMC, their rate of service availability was not so significant in increment, due to relatively similar web services which also perform similar nonfunctional qualities. This can be assumed to be the result of similar recurring service requests which exist within the cloud. The central server PC is stabilized and the feedback records are properly aligned to enhance relatively optimal selection without excessive delay of the queries.

In a similar manner, the throughput measures the rate of successful service request delivery over the AMC communication channel. It determines the time it takes to perform a transaction (E) over a number of sessions. The throughput was measured through the JMeter simulator that was integrated via the Netbeans IDE tool. Since the proposed AMC system consists of connecting mobile devices, the experiment is vital to access the performance of the service selection mechanism. To compute the throughput of the system, the following expression is used:

$$\frac{1}{k} \sum_{c=0}^{k} ET, \tag{16}$$

where the variable c represents the request index and k is the number of requests at a particular session. Figure 7 shows the results of the throughput for the load generator.

The number of mobile nodes represented the number of available mobile devices within the system at the time the experiment was conducted, while the number of simultaneously issued service requests showed the number of users that were making an HTTP request during the experiment. The throughput of a system clearly showed the number of completed service deliveries to the requesting service consumer. The results displayed on Figure 7 show that the throughput performance of the web service selection mechanism increases as the number of users increases. The proposed selection mechanism performed well when the test for throughput was conducted. The linear increment in the values of the throughput with the increasing number of node requests performs in a realistic way. Thus, the throughput averagely increases with the increasing number of requests, which

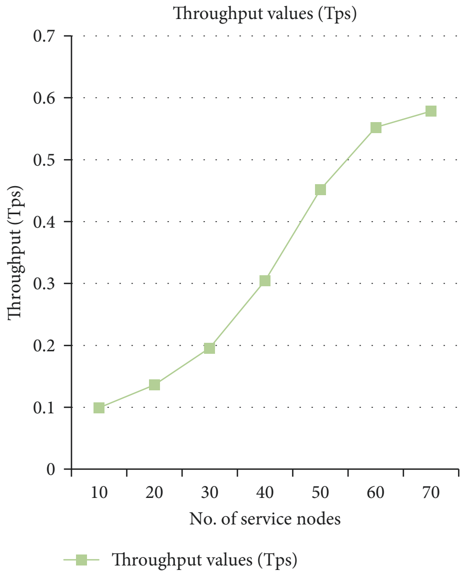

Figure 7: Mechanism selection throughputs.

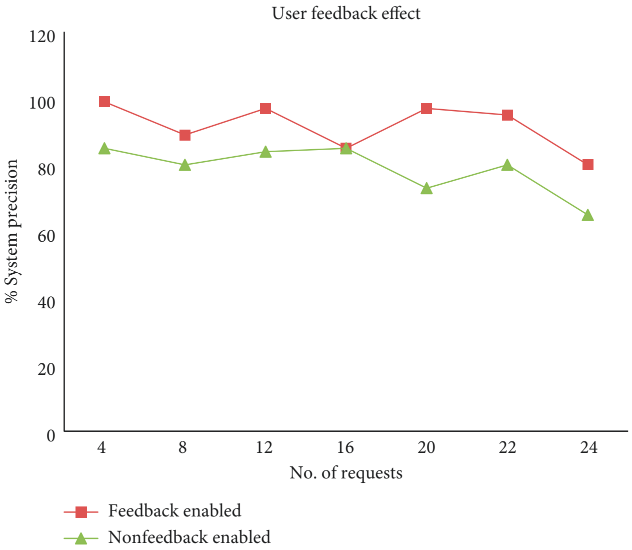

Figure 8: Selection precision based on the feedback effect.

confirms a good performance from the system. Therefore, having confirmed the performance of the AMC system with respect to provisioning of service availability as well as the rate of completed service delivery, we have achieved a good platform upon which the effect of feedback can be tested on the system as well as the contribution of the selection mechanism to the body of knowledge when compared with other AMC selection approaches under similar conditions (occurrence of selection ties).

7.2. User Feedback Effect Evaluation. The proof of the concept system here was implemented through deploying the Glassfish Web Server 3.1.2 platform on the central server PC within the AMC. This platform works together with the SQLite database which acquires the WSDL file as well as the QoS information provided by the various mobile service providers. The test deployed a total of 100 web services for this experiment to simulate a real-world selection challenge in terms of user dissatisfaction. To evaluate the feedback-enabled QoS selection, this experiment evaluated the number of service selections that coincide between feedback-enabled and nonfeedback service selections which were adopted from the existing works [45, 46]. The goal of this experiment is to determine how precise the two approaches are at ensuring that the most relevant service is selected.

In the first approach, the service was selected directly without the aid of feedback ratings of the ad hoc mobile selection mechanism (an approach that is already used in cloud computing), while in the other approach, the selection was carried out with the aid of the feedback selection ratings.

The same values of QoS parameters are requested with the increasing number of users. This evaluation revealed the kind of services retrieved after each service request and recorded the level of satisfaction ranging from a 1-star to a 5-star rating. Starting with the feedback-enabled approach, the expression for percentage precision is given as

$$\text{Precision} = \frac{|\{\text{relevant web services}\} \cap \{\text{retrieved web services}\}|}{|\{\text{retrieved web services}\}|} \%. \tag{17}$$

This gives the percentage precision through calculating the mean of responses from the number of users at each level of the service request. This precision is expressed to determine the percentage level of satisfaction derived by each of the service users. Figure 8 shows that the feedback indicates highly satisfactory service selection at every incidence of service invocation when compared with selection based on the quality of service alone. At some point during selection ties, the nonfeedback effect can make a selection that coincides with the optimal service at that point, and this accounted for both approaches achieving a similar value only at experiment number four where the number of requests is 16. In addition, the feedback-enabled selection mechanism enhanced the selection of web services and also gave a high tendency to solve the issue of service selection ties, thereby providing an optimal service for users.

The user feedback effect experiment shows that the use of a continuous updated and unlimited range of users' web service assessment only enhances the selection of optimal services for the service users. It should, however, be noted that the greater the number of services that attain the selection ties, the higher the effect of the feedback-enabled system in selecting appropriate and more satisfying services for the user. In a situation where there were few numbers of services, both approaches often, though not always, assumed the same output result. The feedback rating system, however, enhanced the selection of the highly rated service

Table 4: QoS metrics for various available e-mail verification web services.

Web services	Response time (ms)	Throughput (b/s)	Availability (%)	Cost (rands)	Reliability (%)	Feedback (%)
XMLLogic	720	6	85	1.2	87	59
XWebservices	1100	1.74	81	1	79	49
StrikeIron Emails	710	12	98	1	96	86
CDYNE	912	11	90	2	91	65
Webservicex	910	4	87	0	83	89
ServiceObjects	1232	9	99	5	99	94
StrikeIron Address	391	10	96	7	94	70
Kickbox	428	5	86	8	60	67
Byteplant	601	8	70	4	75	100
QuickEmail	205	3.5	95	5	89	62
TowerData	504	9	75	1	77	79
Leadspend	832	8	81	3	83	76
Briteverify	911	3	80	5.1	78	26
Mailbox	604	10	89	5	87	69
Emailanswers	505	7	85	6	95	29
BulkEmailVerifier	600	5	90	4	88	96
BulkEmailChecker	195	3.5	85	0	89	55
Emailtor	220	6	87	8	75	28
Verifalia	950	5	90	3	86	38
Xverify	350	8	96	4	94	56

among the selection ties for the user at each point of service invocation where ever it occurs.

7.3. Feedback Mechanism on Real-Life Scenario. This experiment extracted data samples from the work of Al-Masri and Mahmoud in [45]. This gives us the QoS of seven different web services that were chosen from the same domain. These web services are e-mail web services that share the same functionalities. The data were provided in StrikeIron.com and XMethods.net. A total number of 20 web services were used in this experiment to actually see the effect of the proposed selection mechanism at solving the occurrence of web service ties. However, if more web services are deployed, then the possibility of web service ties increases in the system.

The QoS of the 20 web services is measured by WS-QoSMan. WS-QoSMan is a module that is responsible for collecting and measuring the QoS information of web services. The values of the QoS metrics are normalized and assigned weights accordingly as described in Section 3. The free spider simulator (FSS) tool is also an online tool that helps to get a free report about web service actionable insights. The feedback for over 2 weeks was collected and presented. The last column of Table 4 also specifies the feedback of each of the web services as derived from the FSS. The results of various web services were computed as shown in Table 4. We compare our approach with the two other approaches used in the literature which are the web service ranking function approach [45] and nonfeedback outputs from [47]. The values of the data collected under each approach were expressed in Table 5 which are the normalized and ranked outputs. More details can be found in [48].

7.4. Discussions. The results from Table 5 and Figure 9 showed a critical observation that both feedback and nonfeedback techniques achieved the same ordering of web services according to QoS-based selection although different values were obtained and the same ordering was realized. This confirmed that both techniques achieved similar results. However, the WsRF technique only selects at random from the set of services that attain similar QoS scores. The idea discussed in Section 4 helps to solve the challenge by carrying out the prioritization based on the trusted user feedback data that were generated by the AMC web service selection mechanism.

The e-mail validation web services of numbers 6, 9, and 16 (“ServiceObjects,” “Byteplant,” and “BulkEmailVerifier,” resp.) all achieved similar QoS computation. It is understood however that the normalization technique found in [31, 40, 42] ensures that it provides the three best web services amongst those considered. However, this information is not enough to justify the best of the three web services as the feedback rating system shows quite clearly that the web service number 9 (Byteplant) is rated above others of comparable performance. The rating of the Byteplant e-mail validation was seen to be relatively constant over a two-week period and maintained a rating of 100% from different service users. This final choice of the Byteplant over the other two is not only confirmed by the high range of computation selected based on the normalization during QoS computation but

Table 5: Results of WsRF, non-FB, and FB QoS metric computation.

ID	Web services	WsRF computation	Rank	Nonfeedback-enabled computation	Feedback-enabled computation	Rank
1	XMLLogic	3.6638	13	4.7648	0.68	13
2	XWebservices	3.2166	15	4.3176	0.64	16
3	StrikeIron Emails	4.6103	4	5.7113	0.87	5
4	CDYNE	3.9246	11	5.0256	0.72	11
5	Webservicex	4.2679	5	5.3689	0.89	4
6	ServiceObjects	4.6700	1	5.7705	0.90	3
7	StrikeIron Address	4.1955	7	5.2965	0.79	8
8	Kickbox	4.0428	10	5.1438	0.74	10
9	Byteplant	4.6700	1	5.7704	0.97	1
10	QuickEmail	3.9205	12	5.0215	0.69	12
11	TowerData	4.2504	6	5.3514	0.83	6
12	Leadspend	4.0832	8	5.1842	0.81	7
13	Briteverify	2.0911	20	3.1921	0.58	20
14	Mailbox	4.0604	9	5.1614	0.75	9
15	Emailanswers	3.0505	18	4.1617	0.62	18
16	BulkEmailVerifier	4.6700	1	5.7706	0.92	2
17	BulkEmailChecker	3.2195	16	4.3205	0.65	15
18	Emailtor	2.3020	19	3.4031	0.60	19
19	Verifalia	3.0950	17	4.1961	0.63	17
20	Xverify	3.5077	14	4.6087	0.66	14

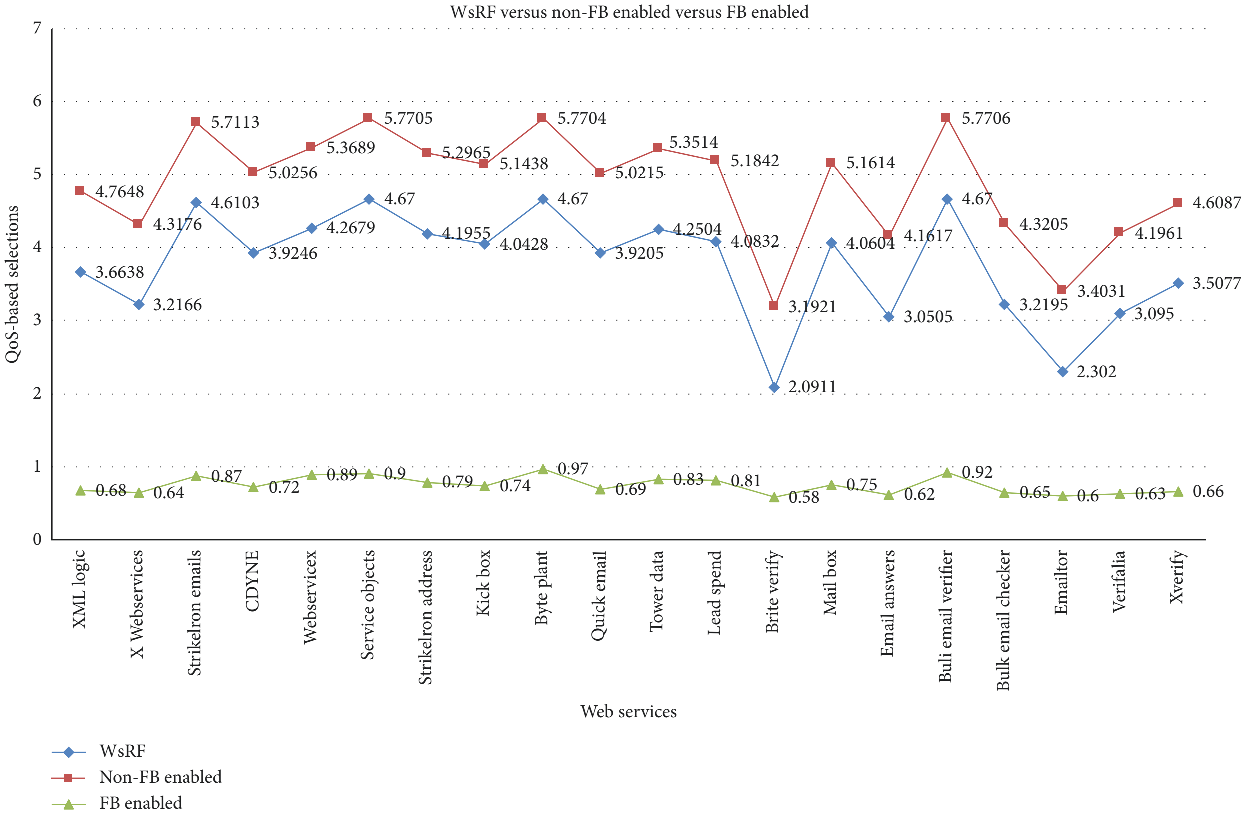

Figure 9: The comparison of WsRF, non-FB, and FB-enabled web service selection.

also affirmed by the extensive feedback range of quality maintenance. The binomial probability distribution used by the recommendation system predicts in such a manner that web services are arranged orderly even to the least significant value to bring about the needed differences to differentiate a web service from another one of similar computation. Hence, the whole experiments were focused on improving the user satisfaction in the AMC, as this will promote frequent patronage and regular interest in consumption of services on the GUIISET business platform.

Summarily, multidynamic and feedback mechanisms have contributed to the knowledge base of AMC service selection in the following ways:

(1) Outrightly removal of the possibility of delay/redundancy as a result of selection ties

(2) Breaking of service selection ties whenever they occur

(3) Ensuring that an optimal service is selected in any service requisition

(4) Promotes prompt service delivery via the multidynamic distance approach

(5) Increasing the user satisfaction level in the course of service provisioning.

8. Conclusion and Future Work

Many services in the AMC platform are with similar functional qualities which open up the understanding that the effective and efficient use of a various combination of nonfunctional QoS will enhance higher user satisfaction through provision of optimal services. We performed a series of experiments within interconnected mobile network nodes as well as P2P mobile devices. Our experiments proved that distances within mobile networks can be an important QoS tool that will enhance the optimal selection process, which is typically useful in the case of emergency needs, for example, natural disaster, mining process, and health monitoring systems.

Moreover, the user feedback effect on web services can be computed to effect proper ordering of these services within the mobile devices in such a manner that eradicates selection ties. The outcome of the experiments showed that, out of seven selection samples, only one is likely to produce an optimal selection without the feedback mechanism with the calculated probability of 0.13. Thus, we can conclude that the use of the feedback mechanism section of the proposed system enhances better performance in users' satisfaction in comparison with existing works such as WsRF [45] and nonfeedback [47].

Possible future works need to consider the context of the device and mobile environment. Thus, inculcating the environmental context as well as the device context into the selection process is a good note for the furtherance of this work with proper checks in place to avoid complexity issues. Another area for future work includes the implementation of an autoswitching system into the mechanism to switch to the new server during failure.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work is based on the research support in part by the National Research Foundation of South Africa (Grant UID: tp11062500001 (2017-2018)). The authors also acknowledge funds received from the industrial partners Telkom SA Ltd., Huawei Technologies SA (PTY), and Dynatech Information Systems, South Africa, in support of this research.

References

[1] R. Yu, X. Yang, J. Huang, Q. Duan, Y. Ma, and Y. Tanaka, "QoS-aware service selection in virtualization-based cloud computing," in *Proceedings of 14th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 1–8, Seoul, Republic of Korea, September 2012.

[2] M. Sathya, M. Swarnamugi, P. Dhavachelvan, and G. Sureshkumar, "Evaluation of QoS based web-service selection techniques for service composition," *International Journal of Software Engineering*, vol. 1, no. 5, pp. 73–90, 2012.

[3] S. Marston, Z. Li, S. Bandyopadhyay, J. Zhang, and A. Ghalsasi, "Cloud computing—the business perspective," *Decision Support Systems*, vol. 51, no. 1, pp. 176–189, 2011.

[4] G. Zou, Q. Lu, Y. Chen, R. Huang, Y. Xu, and Y. Xiang, "QoS-aware dynamic composition of web services using numerical temporal planning," *IEEE Transactions on Services Computing*, vol. 5, no. 3, pp. 1–14, 2012.

[5] B. Martini, F. Paganelli, A. A. Mohammed, M. Gharbaoui, A. Sgambelluri, and P. Castoldi, "SDN controller for context-aware data delivery in dynamic service chaining," in *Proceedings of 1st IEEE Conference on Network Softwarization: Software-Defined Infrastructures for Networks, Clouds, IoT and Services, NETSOFT*, pp. 1–5, London, UK, April 2015.

[6] E. E. Marinelli, "Hyrax: cloud computing on mobile devices using MapReduce," vol. 389Pittsburgh, PA, USACarnegie Mellon University, September, 2009 M.S. thesis, Thesis.

[7] F. AlShahwan and M. Faisal, "Mobile cloud computing for providing complex mobile web services," in *Proceedings of 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, pp. 77–84, Oxford, UK, April 2014.

[8] N. Kaushik and J. Kumar, "A literature survey on mobile cloud computing: open issues and future directions," *International Journal of Engineering and Computer Science*, vol. 3, no. 5, pp. 6165–6172, 2014.

[9] K. Bahwaireth and L. Tawalbeh, "Cooperative models in cloud and mobile cloud computing," in *Proceedings of 23rd International Conference on Telecommunications, (ICT 2016)*, pp. 1–4, Thessaloniki, Greece, May 2016.

[10] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proceedings of ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*, pp. 35–39, San Francisco, CA, USA, June 2010.

[11] A. T. Akinola, M. O. Adigun, A. O. Akingbesote, and I. N. Mba, "Optimal service selection in ad-hoc mobile market based on multi-dynamic decision algorithms," in *Proceedings of 2nd World Symposium on Computer Networks and Information Security*, pp. 1–6, Hammamet, Tunisia, 2015.

[12] A. T. Akinola, M. O. Adigun, A. O. Akingbesote, and I. N. Mba, “Optimal route service selection in ad-hoc mobile E-marketplaces with dynamic programming algorithm using TSP approach,” in *Proceedings of International Conference on E-Learning Engineering and Computer Softwares*, pp. 74–81, Kualar Lampur, Malaysia, 2015.

[13] J. Cao, K. Hwang, K. Li, and A. Y. Zomaya, “Optimal multiserver configuration for profit maximization in cloud computing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1087–1096, 2013.

[14] M. E. Buthelezi, M. O. Adigun, O. O. Ekabua, and J. S. Iyilade, “Accounting, pricing and charging service models for a GUISET grid-based service provisioning environment,” in *Proceedings of the 2008 International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government, (EEE 2008)*, pp. 350–355, Las Vegas, NV, USA, July 2008.

[15] O. O. Ekabua and M. O. Adigun, “GUISET LogOn: design and implementation of GUISET-driven authorization,” in *Proceedings of First International Conference on Cloud Computing, GRIDs, and Virtualization 2010*, pp. 1–6, Lisbon, Portugal, November 2010.

[16] E. K. Olatunji, M. O. Adigun, E. Jembere, J. Oladosu, and P. Tarwire, “A privacy-as-a-service model for securing data in GUISET environment,” in *Proceedings of Southern Africa Telecommunication Networks and Applications Conference (SATNAC’13)*, pp. 143–148, Stellenbosch, South Africa, September 2013.

[17] A. O. Akingbesote, M. O. Adigun, S. Xulu, and E. Jembere, “Performance modeling of proposed GUISET middleware for mobile healthcare services in E-marketplaces,” *Journal of Applied Mathematics*, vol. 2014, Article ID 248293, 9 pages, 2014.

[18] A. O. Akingbesote, M. O. Adigun, J. B. Oladosu, and E. Jembere, “A quality of service aware multi-level strategy for selection of optimal web service,” in *Proceedings of IEEE International Conference on Adaptive Science and Technology ICAST*, pp. 1–6, Pretoria, South Africa, November 2013.

[19] A. O. Akingbesote, M. O. Adigun, M. A. Othman, and I. R. Ajayi, “Determination of optimal service level in cloud E-marketplaces based on service offering delay,” in *Proceedings of 2014 IEEE International Conference on Computer, Communication and Control Technology (I4CT 2014)*, pp. 283–288, Langkwawi, Kedah, Malaysia, September 2014.

[20] T. Shezi, E. Jembere, and M. Adigun, “Towards developing failure tolerant communication framework for GUISET services,” in *Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNACT’11), East London International Convection Centre*, East London, South Africa, 2011.

[21] A. O. Akingbesote, M. O. Adigun, E. Jembere, and J. Oladosu, “Modeling the cloud e-marketplaces for cost minimization using queuing mode,” *Australian Journal of Basic and Applied Sciences*, vol. 8, no. 4, pp. 59–67, 2014.

[22] A. O. Akingbesote, M. O. Adigun, S. S. Xulu, M. Sanjay, and I. R. Ajayi, “Performance analysis of non-preemptive priority with application to cloud E-marketplaces,” in *Proceedings of IEEE International Conference on Adative Technolgy (ICAST)*, pp. 1–6, Paris, France, 2014.

[23] S. Msane, M. O. Adigun, and O. O. Ekabua, “A reputation-based trust management model to enforce trust amongst web services in a GUISET grid environment,” in *Proceedings of International conference on Semantic Web abd Web Services (SWWS’08)*, pp. 230–235, Las Vegas, NV, USA, July 2008.

[24] M. Swarnamugi, “Taxonomy of web service selection approaches,” *International Journal of Computer Applications*, vol. 2, no. 4, pp. 18–22, 2013.

[25] M. Klusch and P. Kapahnke, “Semantic web service selection with SAWSDL-MX,” in *Proceedings of Second International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMRR 2008)*, pp. 1–15, Karlsruhe, Germany, October 2008.

[26] S. Susila, S. Vadivel, and A. Julka, “Broker architecture for web service selection using SOAPUI,” in *Proceedings of International Conference on Cloud Computing Technologies, Applications and Management (ICCCTAM 2012)*, pp. 219–222, Dubai, UAE, December 2012.

[27] M. Xin, M. Lu, and R. Zhang, “A location-based services selection model and algorithm with QoS constraints under mobile internet environment,” in *Proceedings of International Conference on Service Sciences*, pp. 124–129, Jiangsu, China, May 2014.

[28] M. Amoretti, M. C. Laghi, A. Carubelli, F. Zanichelli, and G. Conte, “Reputation-based service selection in a peer-to-peer mobile environment,” in *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*, pp. 1–8, Newport Beach, CA, USA, June 2008.

[29] M. Keidl and A. Kemper, “Towards context-aware adaptable web services,” in *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, ACM*, pp. 55–65, New York, NY, USA, May 2004.

[30] N. Keskes, “Context of qos in web service selection,” *American Journal of Engineering Research*, vol. 2, no. 4, pp. 120–126, 2013.

[31] Y. Liu, A. H. H. Ngu, and L. Z. Zeng, “QoS computation and policing in dynamic web service selection,” in *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters, ACM*, pp. 66–73, New York, NY, USA, May 2004.

[32] H. Shah-Hosseini, “Problem solving by intelligent water drops,” in *Proceedings of IEEE Congress on Evolutionary Computation*, pp. 3226–3231, Singapore, September 2007.

[33] A. T. Akinola and M. O. Adigun, “Feedback-based service selection in ad-hoc mobile cloud computing,” in *Proceedings of 3rd International Conference on Advances in Computing and Communication Engineering (IEEE-ICACCE-16)*, pp. 72–77, Durban, South Africa, November 2016.

[34] Z. Zhang, W. Sun, W. Chen, and B. Peng, “An integrated approach to service selection in mobile ad hoc networks,” in *Proceedings of 4th IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WICOM 2008)*, pp. 1–4, Dalian, China, October 2008.

[35] A. T. Akinola, M. O. Adigun, and P. Mudali, “A federated web service selection approach for ad-hoc mobile cloud computing,” in *Proceedings of 19th Southern African Telecommunication Networks and Applications Conference (SATNAC)*, pp. 386–373, Frankhurt, South Africa, September 2016.

[36] A. Varshavsky, B. Reid, and E. de Lara, “A cross-layer approach to service discovery and selection in MANETs,” in *Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, pp. 66–73, Washington, DC, USA, November 2005.

[37] R. Lacuesta, J. Lloret, S. Sendra, and L. Peñalver, “Spontaneous ad hoc mobile cloud computing network,” *Scientific World Journal*, vol. 14, no. 10, pp. 1–19, 2014.

[38] K. Yang, A. Galis, and H.-H. Chen, “QoS-aware service selection algorithms for pervasive service composition in mobile

wireless environments," *Mobile Networks and Applications*, vol. 15, no. 4, pp. 488–501, 2010.

[39] E. Cavalcante, T. Batista, F. Lopes et al., "Optimizing services selection in a cloud multiplatform scenario," in *Proceedings of the IEEE Latin America Conference on Cloud Computing and Communications, LatinCloud*, pp. 31–36, Porto Alegre, Brazil, November 2012.

[40] M. Makhlughian, S. M. Hashemi, Y. Rastegari, and E. Pejman, "Web service selection based on ranking of QoS using associative classification," *International Journal on Web Service Computing*, vol. 3, no. 1, pp. 1–14, 2012.

[41] J. Zhu, Y. Kang, Z. Zheng, and M. R. Lyu, "A clustering-based QoS prediction approach for web service recommendation," in *Proceedings of 15th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops*, pp. 93–98, Shenzhen, China, April 2012.

[42] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnanam, and H. Chang, "QoS-aware middleware for web services composition," *IEEE Transactions on Software Engineering*, vol. 30, no. 5, pp. 311–327, 2004.

[43] N. Kokash and V. D'Andrea, "Evaluating quality of web services: a risk-driven approach," in *Proceedings of Business Information System Conference*, pp. 180–194, Poznan, Poland, April 2007.

[44] O. Kondratyeva, N. Kushik, A. Cavalli, and N. Yevtushenko, "Evaluating quality of web services: a short survey," in *Proceedings of IEEE 20th International Conference on Web Services*, pp. 587–594, Santa Clara, CA, USA, June 2013.

[45] E. Al-Masri and Q. H. Mahmoud, "QoS-based discovery and ranking of web services," in *Proceedings of 16th International Conference on Computer Communications and Networks*, pp. 529–534, Honolulu, HI, USA, August 2007.

[46] R. P. Singh and K. K. Pattanaik, "An approach to composite QoS parameter based web service selection," *Procedia Computer Science*, vol. 19, no. 2, pp. 470–477, 2013.

[47] S. Reiff-Marganiec, H. Q. Yu, and M. Tilly, "Service selection based on non-functional properties," *Service-Oriented Computing-ICSOC 2007 Workshops*, pp. 128–138, Springer, Berlin, Germany, 2009.

[48] A. Akinola, "QoS-based web service selection mechanism for ad-hoc mobile cloud computing," University of Zululand, KwaDlangezwa, South AfricaUniversity of Zululand, 2017 M.Sc. dissertation, Thesis.

Journal of Engineering
The Scientific World Journal
International Journal of Rotating Machinery
Journal of Sensors
Advances in Multimedia
Advances in Civil Engineering
Journal of Control Science and Engineering
Journal of Robotics
Journal of Electrical and Computer Engineering
Advances in OptoElectronics
VLSI Design
International Journal of Navigation and Observation
Modelling & Simulation in Engineering
International Journal of Aerospace Engineering
International Journal of Chemical Engineering
International Journal of Antennas and Propagation
Active and Passive Electronic Components
Shock and Vibration
Advances in Acoustics and Vibration