Journal of King Saud University – Computer and Information Sciences 35 (2023) 209–223

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Service selection model based on user intention and context

Ako A. Jaafar [a,b,*], Dayang N.A. Jawawi [a], Mohd Adham Isa [a], Nor Azizah Saadon [a]

[a] *School of Computing, University Teknologi Malaysia, 81310 Skudai, Johor, Malaysia*
[b] *Software Engineering Department, Qiawan International University, Kurdistan Region, Iraq*

A R T I C L E I N F O

Article history:
Received 20 October 2022
Revised 5 March 2023
Accepted 24 March 2023
Available online 31 March 2023

Keyword:
Service selection
User preference
Internet of Things
User intention and context
Entropy
Skyline services

A B S T R A C T

The internet interconnects billions of objects and services. Real-world services are offered by entities with various functions communicating with one another. The selection of services based on their functionality is a complex process because as the number of services rises, so does the number of services that offer the same functionality. Quality of service (QoS) can be a criterion for selecting a suitable service. However, QoS' relative significance fluctuates due to changing user preferences, and users may exhibit various behaviors depending on their contexts and intentions. Defining a user's preference based on user intentions and context is similarly challenging; nonetheless, scholars have paid little attention to this topic. This study provides a new model for service selection based on user intentions and context. The model dynamically selects the appropriate set of QoS with their importance to specify a user preference for various behaviors. The issue of assessing user preference to select the desired service is resolved by calculating the QoS importance based on the user's behavior history and context. The study proposed a dynamic K-Skyline method to optimize a search space and a multi-criteria decision-making technique to select and rank services efficiently. A case study and an experiment demonstrating the proposed model are presented, in which real-world datasets are utilized. The experimental results of the proposed model validate the model's efficiency and robustness.

1. Introduction

Web service is an interoperable software system that facilitates machine-to-machine communication via the internet to provide services. It is a type of distributed computing that enables access to various computational resources through the use of industry-standard protocols such as HTTP, which guarantees compatibility with various devices and formats. With an increase in the number of internet-based services, the number of services that provide a similar functionality has also expanded. This trend accelerated with the advent of the Internet of Things (IoT) and cloud computing. Based on IoT's vast protentional, various applications have been developed such as smart cities, smart healthcare, smart agriculture, and smart education (Jaafar, 2019). The increasing number of smart devices has led to an increase in the number of functionally identical services that can be used to complete an activity, each with a different Quality of Service (QoS).

Service discovery and selection are becoming challenging and time-consuming tasks. The fact of having a high number of services that provide similar functionality drives many studies to concentrate on service discovery and selection based on various approaches such as considering QoS, service context, user context, preferences, and intentions.

The purpose of the service selection model is to select appropriate services that meet the needs of customers. Gathering the requestor's desires, evaluating available services, aggregating the assessment results, ranking, weighting the top returned services, and finally picking the best service for the consumer are all part of the service selection model (Yu, (n.d.)). Several variables must be considered while selecting a service, including precondition, effect, QoS, business rule, and policy. Selecting the proper service with multi QoS and different contexts for each service is a challenging task. The task becomes more challenging as the number of alternative services is increasing by adopting services offered by SOA, cloud computing, and IoT.

According to the existing research, service selection for various types of services, such as traditional web services, cloud services, micro services, and IoT services, has been considered by many

* Corresponding author.
E-mail addresses: ajako2@live.utm.my, ako.abubakr@uniq.edu.iq (A.A. Jaafar).

Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2023.03.018
1319-1578/

researchers over the last two decades. The studies focused on a variety of service selection factors, such as service description, in order to locate and pick the desired service by matching the request to the description of the service. Describing QoS service, service context, and user context is another aspect of enhancing service matching through further description. The primary focus of QoS-aware and context-aware service selection is on making the best possible QoS and contextual decisions for each user and each service. Many theories and methods have been put out to address the issue of selecting services with consideration for QoS and context. Researchers have exerted significant effort to determine the optimal method for selecting a service based on the QoS provided by either directly from the service provider or inferred from past experiences. To determine the most convenient service for the user, numerous researchers incorporate user preferences into service selection through direct user preference definition.

User preferences are the most essential indicators in service selection and personalized choices, in which, requesters declare their demands in terms of functional and non-functional criteria and their respective weights. User preference determines the weight of the QoS and the context of the required service. Multi-Criteria Decision Making (MCDM), recommendation methods, and context-aware methods are the most popular methods and techniques to utilize user preferences.

The most used MCDM approach in service selection domains are Analytical Hierarchy Process (AHP), Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), VlseKriterijuska Optimizacija I Komoromisno Resenje (VIKOR), and the Bad Worth Method, have been applied (BWM). These methods prioritize the alternative choices for the user based on user preferences and weights. These methods are biased when they depend on the supplier's set QoS criteria with expert opinions, however, users cannot assess the QoS based on the performance metrics that are provided by the service provider only (Wang, 2020).

The level of QoS importance will be determined by its given weight (Singh et al., 2020), to estimate the weight, in certain studies, requesters are forced to express their preferences for each QoS. This is a demanding task that may result in undesirable service delivery (Benouaret et al., 2012). In addition, from the perspective of the service provider, offering high-quality QoS can greatly improve its business competitiveness (Liu et al., 2004). Nevertheless, relying on QoS criteria presents numerous challenges. One of the challenges is the importance of QoS and how users express their preferences for diverse domains, intentions, and behaviors. Accordingly, in the last decades, user preference consideration has been the subject of numerous researches for service selection (Wang, 2020; Zhao, et al., 2017; Alaoui et al., 2015; Najar et al., 2015; Daosabah, et al., 2021). Some researchers have focused on direct preference expression by a user, while MCDM algorithms are considered to find user preference through users or expert opinion. Recommendation systems and user similarity are two other user-centric approaches that are widely utilized in the service selection domain (Xu, 2016). A few studies have attempted to solve the mentioned problem by specifying user preference and learning about user preference through users' historical service invocation and situation.

Research in the field has demonstrated that individuals with similar contexts, such as geography, invocation time, domain interest, user type, age, and gender, have more similarities than users with dissimilar contexts (Xu, 2016; Qi, et al., 2015; Qi, 2017). Context is any information that can be utilized to describe the entity's state. An entity is a thing or item that is thought to be significant to a user's engagement with an application, including the user and the application itself. Context-aware is a representation of the user's preferences, and it will provide typical services to the user, which may not be adequate for the user's requests. Changing user needs are caused by the evolution of user intentions (Daosabah et al., 2019). Past researches have shown that users with comparable contexts such as location, invocation time, domain interest, user type, age, gender, etc. have more in common with one another than users with dissimilar contexts (Xu, 2016; Qi, et al., 2015; Qi, 2017). In addition, the situation in which an intention arises has a significant bearing on its fulfillment. Different services can be suggested based on the environment. This means that the same goal can be met in different ways depending on the context (Najar et al., 2015; Daosabah, et al., 2021). From that vantage point, many researchers have utilized user and service context in user-centric service selection. Additionally, researchers have studied the influence of user intention on selecting proper services (Alaoui et al., 2015; Najar et al., 2015; Daosabah, et al., 2021) and they have demonstrated that the change in user intent drives a shift in users' needs. Hence, the context and intent of a user's behavior have a direct impact on their preference.

An intention is a statement that expresses a state that is anticipated to be attained or maintained. It can also be defined as an objective to be attained by executing a sequence of intentions and methods for the target intention (Alaoui et al., 2015). Intentions are a high-level explanation of the user's objectives, outlining their requirements for the service and demonstrating why they need it, without detailing how these criteria could be met (Najar et al., 2015). Researchers have emphasized user intent primarily for user preference learning in recommendation systems. The user's preference is unaffected by the circumstances in which a recommendation is delivered, even though the user's intent may vary every time. From this vantage point, researchers aim to identify user intent for use in service discovery, selection, and composition. Furthermore, the importance of QoS may vary between different users, decision-makers, and different user's behavior. However, preference and QoS importance variations based on distinct intentions and behaviors have not been considered.

According to the extent of our knowledge, no study identifies and specifies user preferences based on how users act in response to their intents in a particular context. This paper proposes a new service selection model based on user intention and context to define automatic user preferences. The main aims of this paper are to propose an Intention and Context-based Service Selection Model (ICSSM) and a new method to calculate user preference based on user behavior. Moreover, a dynamic K-skyline algorithm and a TOPSIS are proposed to optimize and rank the final selection.

1.1. Motivation

User preference is highly significant in selecting services to prefer a service where they published to accomplish users' goal. According to the background study, a lot of researchers studied user preference in service discovery, selection, and composition. Since selecting the best service among a lot of service providers is challenging, user preference makes the selection more suitable for the user. While calculating user preference is challenging, when it comes to calculating automatic user preference, considering the changes in the preference for different domains and behaviors, it is even more challenging.

In another hand with the adoption of IoT and smart cities, the number of service providers and service consumers has increased dramatically, and in the future, every human or thing may consume hundreds of different services every day in a smart environment. In 2030, Cisco predicts, there will be 500 billion things connected to the Internet. According to study (Kaur et al., 2018); the cities of Dubai and Abu Dhabi are introducing 250,000 smart meters and 5,000 Wi-Fi hotspots to provide free internet to essential sectors such as healthcare, education, safety, telecommunications, tourism, and utilities. In addition, India aims to construct more than one hundred smart cities, while Barcelona is introducing e-government and contactless services. This significant increase in the number of connected devices results in an increase in the number of service providers and consumers, while these devices may operate as either service providers or service consumers, or both. Consequently, establishing user preferences becomes more difficult, especially when such choices alter in response to shifting behaviors and circumstances. There is no unique set of QoS attributes that will be provided to users of all service providers and all domains, which is an additional difficulty with user preferences. Users must therefore define distinct preferences for service selection in each domain. The suggested model selects the suitable service for the user based on their predicted preferences, which are derived from their intentions and context.

1.2. Contribution

This study presents a model for service selection based on user intent and context. To predict a user preference, the proposed model uses historical user intentions achieved in a specific context. The proposed model employs the historical users' intention to predict a user preference. The proposed model recognized the behaviors to differentiate preferences for the same intention in different behaviors. The main contributions of this study are.

- Propose a new service selection model to select services using predicted users' preferences based on their intention and context.
- The generation of behavioral similarity by combining semantic, Jacquard, and sequence similarity.
- Propose a dynamic K-Skyline algorithm for optimizing the service selection search space.

1.3. Organization

The rest of this paper is structured as follows: Section 2, provides the Background for the Research and examines a number of related works. In section 3, the proposed service selection model and the research methodology followed in this study are described. Section 4, is a case study to demonstrate the application of the proposed method. In section 5, experimental analyses are presented to test the efficiency and robustness of the proposed model. Section 6 presents a discussion of the study and its contribution. At the end, the conclusion, limitation, and future direction of this study are presented.

2. Related work

In this session, the related work of this study is discussed from three separate aspects: user preference and context, intention, user behavior and intention, and service selection mode.

2.1. User preference and context

A large number of scholars have acknowledged and focused on the critical role that user preferences play in the selection of services, and as a result, many different facets of gathering consumer preferences have been investigated. The most wanted IoT service was determined by combining the preferences of decision makers with the proposed MCDM approach, as shown in study (Baranwal et al., 2020). Decision-makers' preferences are expressed through a new framework developed in the paper, which combines an OWA (ordered weighted averaging aggregation) operator with Fuzzy TOPSIS. The proposed model in study (Jin, 2016) takes into account user preference for calculating the absolute dominance of IoT services among available services.

To get around the user's restricted options when requesting a service, the study (Zhao, et al., 2017) proposed automatically learning user preferences for service selection and composition. The proposed model chooses services based on similar user preferences and profiles, such as the user's hobbies or educational background. In order to mine customers' preferences from their scoring data, the study (Wang, 2020) presented an AP and K-mean clustering technique, where users with similar ratings for the same service are placed together into the same cluster. Another study proposed a new semantic clustering approach based on the lattice concept, which clusters web services based on their semantic similarity. The proposed approach utilizes user preference and user context to generate sub lattice from the cluster service to minimize the semantic differences between the requested service and discovered services (Natarajan, 2020).

Service rating has been proposed by many researchers which relies on customer rating or frequency of successful service invocation. These approaches also suffer from two problems. First: the model could not resist malicious rate attacks. Second: the model will not be able to predict a rating for a node if nobody has rated it yet; this is an inherent problem that occurs, especially in decentralized service selection (Kanagaraju and Nallusamy, 2018; Nwe et al., 2014). Additionally, the amount of QoS criteria varies by domain, this makes it extremely difficult to evaluate each QoS for each domain and service, particularly for IoT services in which the service requester could be a person or a device. Consequently, user preference must reflect genuine user desire and must be predicted without user intervention to respond to the highly dynamic environment problem as well as the fluctuation of the user's behavior toward their intentions in specific contexts.

Moreover, numerous research efforts have been dedicated to exploring the concept of selecting services while taking context into account. In (Qi, et al., 2015) context-aware service selection is presented, which makes service recommendations by relating user context to service context (such as location). Many contextual data of the users and device are employed in service selection, such as age and location for human users and device information for the requester device, where context similarity greatly affects service selection (Najar et al., 2015; Daosabah, et al., 2021). The LISA framework (Gochhayat, 2019) utilizes user context, such as location, time, age, educational background, and economic status, to automatically generate user queries and select relevant services through adapting service profiles to user context and profile.

2.2. User behavior and intention

Various studies attempted to mine and capture user behavior through user context and situations. The study (Ali et al., 2017) proposed a user behavior-based service selection to capture user preference and situation in a smart environment to select IoT services. In (Najar et al., 2015; Najar et al., 2014), the authors present a new method for service discovery and recommendation based on user context and purpose, with intent presented as a verb and target. The verb specifies the activity, enabling the interpretation of intent. The target is either the pre-existing object or the consequence of the action that enables the verb to be carried out. When a user makes a service request, they do so with a specific goal in mind, and the service should be tailored to meet that goal, as per the intentional perspective. Consequently, a user-centric perspective of service orientation has evolved, which considers the user's intention (Najar et al., 2011). Different service qualities influence the user's continued intent to utilize a certain service. The study (Yang and Lin, 2015) demonstrates that the usefulness and privacy protection of a cloud storage service can influence a consumer's choice.

The characteristics of users and their choice of service delivery channels determine their continued intention to use smart city services (Salim, 2020). The article (Daosabah et al., 2019) describes a study on dynamic service compositions based on context and intent. They believe the intent is necessary to accomplish user-centric service composition. The frequent demands of a user cannot be met by user profiles alone; intentional user expression aids in providing the user with the best service. To support this claim, the authors of (Daosabah, et al., 2021) suggest the integration of context and aim in service composition. In their paper, they used semantic descriptions of user goals to determine user intention in the absence of user engagement. To choose the best service for a user, demand frequency has been weighted with time and place and implemented in (Qi, 2017). According to the study, there is a clear link between service selection accuracy and frequency of invocation. Therefore, through the frequent demand and invocation of service, user behavior could be detected, which is established based on multiple intentions in a specific context.

2.3. Service selection model

Numerous studies have proposed and improved models for service selection. In the studies (Zhao, et al., 2017; Gochhayat, 2019) a service selection model is proposed that describes the service description model, user preference, service ranking, and selection algorithm. In their approach, they use three different models to describe IoT services: service, entity or device, and resource based on OWLS. The physical aspects of IoT services are emphasized in these studies. The authors of (Natarajan, 2020) have suggested a new semantic service selection model based on lattice clustering to cluster web services according to the semantic similarity of service operations. The paper (Gochhayat, 2019) offers LISA, a lightweight context-aware Internet of Things service architecture that efficiently filters and delivers the most important and relevant services to consumers based on their context. LISA creates a user model that uses agents and available web services to resolve local decision-making. The suggested user model abstractly characterizes the user by considering the user's context and profile information. Furthermore, the proposed model contains a user preference model for analyzing user inquiries on two levels: criteria qualification and criteria quantification.

Moreover, a lightweight service description has been considered for the service selection and discovery model to present a better performance, such as WSMO-Light. WSMO-Lite (Wei, 2016); Three Particular Predicates (TPP) (Xiang, 2015), and LISA for IoT push services (Gochhayat, 2019). The authors of (Natarajan, 2020) have suggested a new semantic service selection model based on lattice clustering to cluster web services according to the semantic similarity of service operations. In addition, the suggested model includes a user preference model to analyze user queries on two levels: qualification of criteria and quantification of criteria.

Numerous academics view service selection as an optimization issue. Many service selection models presented which is focused on optimizing service selection, using heuristics, *meta*-heuristics, and non-heuristic methods. Local and global optimization has been employed in service selection and composition, such as Skyline algorithms (Jin, 2016; Jin, 2014; Abosaif and Hamza, 2020). Non-heuristic approaches, such as graph theory (Nizamkari, 2017), and MCDM approaches, were used by some researchers. MCDM approaches are widely employed for service selection and ranking based on a requester's preferences. The researchers focus on developing a selection model that considers QoS, context, and user preferences for service selections. The study (Baranwal et al., 2020) presented a framework for IoT service selectin based on Fuzzy-TOPSIS and OWA which utilized decision's maker preference, QoS importance, and service ranking. Additionally, a service selection model based on hybrid Skyline–Entropy–Fuzzy-AHP-PROMETHEE (SEFAP) for service selection proposed. The suggested approach makes use of the strengths of existing approaches while improving upon their individual performance.

3. Proposed model

This section of the article focuses on the proposed model for service selection based on user intention and context as a potential solution to the aforementioned issue. The model makes use of the user's intention as well as the context to identify the user's behavior, based on which user's preference to select a service is identified. User behavior is formed by a chain of the user's past intentions and the context in which they were engaging over a period of time. Fig. 1 shows the three key parts of this model which are: Clustering, Pre-selection, and Selection. The figure also presents the components, flow of the data and the service selection process of the proposed model. The clustering component illustrates how user behavior is grouped according to users' intentions and the context in two steps: first, the similarity between intentions and context is computed, and later, user behavior is grouped according to calculated similarities using a clustering algorithm.

The second part of the presented model is called "Pre-Selection", it's responsible for capturing a user's query and preparing g effective QoS with their importance for the service selection. The selection takes place based on what the user has indicated in the most similar behavior considering intention and context. This portion of the model utilizes the clustered behavior established in the preceding phase of clustering. The model concludes with a selection component that selects services in three steps. The initial step entails capturing a user request and matching it with available online services to identify candidate services that provide the same functionality and serve the same user purpose. In the second step, the search space will be optimized based on the effective QoS criterion from the pre-selection phase. The third step is to rate and select the best-suited optimized service for the user. The following subsections cover the model's specifications.

3.1. Clustering

This process clusters the history of intention and context in two steps: first, the similarity between intention and context histories is calculated, and then a similarity matrix is generated. The second step is to apply a clustering algorithm on the most comparable entries to group them.

3.1.1. Similarity calculator

To calculate similarity, the history of user behavior in terms of the user intention, context, and sequence of occurrence is considered. Context plays a vital role in calculating similarities as user's satisfaction increases with the degree to which the observed user's context matches the desired context, (Najar et al., 2015; Daosabah, et al., 2021). Based on that rationale, a similarity matrix N can be generated. N is an $M \times M$ matrix produced by calculating the similarity ratio in a user's behavior history. Each user invocation contains a user context and an intention (verb and target) according to (Najar et al., 2015; Daosabah, et al., 2021). The user's behavior is anticipated by a combination set of user intentions within a period and possessing a specific context.

Users behave differently toward fulfilling a specific goal, algorithm 1 in Fig. 2 is utilized to calculate the similarity between two users' behavior. The algorithm considers two user behaviors

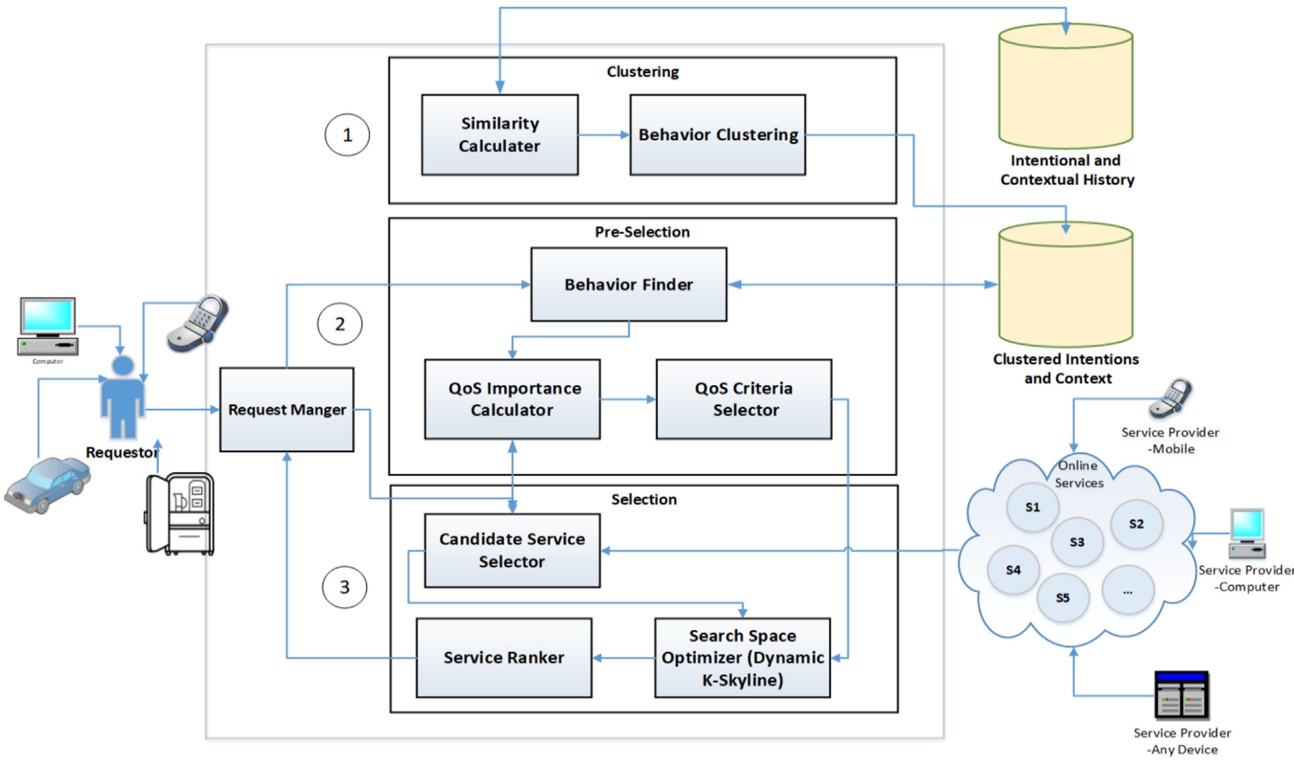

Fig. 1. Intention and Context-based Service Selection Model.

as similar if their context similarity is baggier than the threshold. The context similarity in line eight of the algorithm is semantically calculated based on the semantic similarity in Eq. (1). Each context condition (ctxi) in Eq. (1) is matched with another context condition (ctxj). The semantic calculation of intention similarity is based on a verb and target phrase (Najar et al., 2015), while the similarity of two user behaviors is computed using Eq. (2).

Moreover, the sequence and order of user intention within a period are indicating how user behaves towards selecting services. For instance, if we consider two sets of intentions, set A and set B, and both are intended to invoke services in a particular domain, and both sets consist of three service intentions, it is unlikely that both sets will be executed in the same order such as (GetMap->RentCar->GetEmergency) and (GetEmergency -> RentCar -> GetMap) respectively. Although both sets contain the same set of service intentions, they may have different similarities since they behave differently in terms of the order of the invocation. In set A, GetMap and RentCar precede GetEmergency, which means that the preceding service invocation of GetEmergency may not indicate that they occurred for the same emergency reason, while the GetEmergency service intention appears before RentCar and GetMap in the second scenario, this may indicate that these two intentions are more likely to occur for the same emergency cause. Consequently, the sequence weight will be applied to the similarity of user's behavior. According to Eq. (3), a strong similarity between the occurrences of two behavior will result in a greater weight. Two user behaviors have a high degree of similarity when they share a high degree of semantic context similarity, semantic intention similarity, a similar number of invocations during a period of time, and a similar invocation's sequence, as illustrated by Algorithm 1.

$$ctxSimilarityScore = \sum_{i=0}^{n}\sum_{j=0}^{m} contextConditionMatching(ctxi, ctxj) \quad (1)$$

Where *ctxSimilarityScore* is the total similarity between two user contexts, n is the number of conditions in the source context and m is the number of conditions for the target context.

$$intSimilarity = \sum_{i=0}^{n} w * max(intMatching(inTi, tIlist)) \quad (2)$$

Where intSimilarity is the total similarity of a set of source intentions with a set of target intentions (tIlist), n is the number of intentions in the source intention list, and w is the sequence weight as in Eq. (3).

$$w = 1 - \frac{|i - MaxIndex|}{n} \quad (3)$$

Where MaxIndex is the index of max similarity with I, n is the size of the intention list.

3.1.2. Behavior clustering

Clustering is the method of grouping a collection of items into distinct categories based on their similarities. Clustering has gained a significant amount of attention from scholars in a variety of fields (Wang, 2020). According to the literature, Hierarchical Clustering Algorithm (HCA) and K-Mean clustering are well known for semantic clustering (Eyal Salman, 2018; Purohit and Kumar, 2019). HCA is an analysis technique for clustering that aims to establish a cluster hierarchy. HCA can be performed from the bottom up or from the top down. The bottom-up strategy involves merging two comparable clusters into one until only the root cluster remains. Each stage of the top-down algorithm separates the repository into two groups until no further divisions are available. HCA allows for the use of a variety of linkage criteria. The shortest possible distance between two clusters is utilized by a single linkage. For complete linkage, use the distance between two members of two clusters that are the farthest apart. while the average linkage takes into account the average distance that exists between two clusters (Cong, 2015).

For this study, 500 user behaviors are clustered using three different types of HCA and K-Mean algorithms. Since the clustered data is unlabeled data, the Silhouette method is used to assess the accuracy of the chosen algorithms. The silhouette coefficient

Journal of King Saud University – Computer and Information Sciences 35 (2023) 209–223

```
Algorithm 1: Intention Similarity Algorithm
Input: IntentionListA, IntentionListB
Output: similarity
1   Function getICSimilarity(IntentionListA,IntentionListA)
2   Begin
3           similarity=0
4           UserContexI←getContextof(IntentionListA)
5           UserContexJ←getContextof(IntentionListB)
6           ContexSimilarity←getContextSimilarity(ContexI,ContexJ)
7           IF ContexSimilarity>Threshold
8           Begin
9               TotalLocalSimilarity=0
10              For i=0:IntentionListA
11              Begin
12                  localSimilarity←maxSemanticSimilarity(IntentionListA(i),IntentionListB)
13                  IF(localSimilarity>0)
14                  Begin
15                      sequenceWeight←getSequnceWeight(IntentionListA(i),IntentionListB)
16                      localSimilarity= localSimilarity * sequenceWeight;
17                      TotalLocalSimilarity= TotalLocalSimilarity +localSimilarity
18                  End
19              End
20              IF all IntentionListA has similarities in IntentionListB
21              Begin
22                  intentionSimilarity= TotalLocalSimilarity /((size(IntentionListA)+size(IntentionListB)-
                    TotalLocalSimilarity)
23                  similarity=(1-a)*intentionSimilarity+a*ContexSimilarity
24              End
25              Else
26              Begin
27                  similarity=0
28              End
29          End
30          Return similarity
31  End
```

Fig. 2. Intention Similarity Algorithm.

tests cluster separation regardless of cluster numbers. It's a measure of how close each point in a cluster is to other points in the same cluster compared to points in other clusters. The silhouette range is −1 to 1, a high coefficient means better cluster organization (Rousseeuw, 1987). Accuracy of HCA and K-Means algorithms are compared across a range of cluster sizes. For this analysis, the cluster size will range from 10 to 39. Fig. 3 presents the evaluation results and shows that HCA-A is the best for most cases. In addition, it represents low precision by displaying fewer partitions due to data sparsity. However, in most situations, the HCA-A performs admirably. As a result, HCA-Average Linkage (HCA-A) is employed to cluster user behavior in this study.

The algorithm 1, as shown in Fig. 2, calculates the behavioral similarity. Line number 6 computes the similarity between the source and target user contexts. The line number 15 calculates the sequence weights of two user behaviors. Line 17 computes the intention similarity between any single intention in the source user behavior and the intentions in the target user behavior, and the intention with the highest similarity is chosen as a similar intention while taking their sequence similarity weight into account. For instance, if we assume that there are three user behaviors with the same user's intention but in a different sequence of occurrence, Behavior A: (Get Emergency->Rent Car->Get Map), Behavior B: (Rent Car->Get Emergency->Get Map), and Behavior C, (Get Map->Rent Car->Get Emergency), the similarity of behaviors A and B is higher than the similarity of behaviors A and C since they have higher sequence similarity.

3.2. Pre-Selection

At this stage, the request made by the user will be obtained from the request manager. This request will include the user's intention as well as the context in which they are functioning. The user's intention, in conjunction with its previous intentions in a period of time, constructs the user behavior. Because of this, the captured behavior will be compared to clustered user behavior in order to find the cluster that is the most similar. Subsequently, the weighting approach will be applied to QoS records of users whose behaviors are most similar in order to calculate the QoS importance of the current intention. Finally, the affective criteria for service selection are chosen by the QoS selector.

3.2.1. Behavior finder

On the basis of the requester's behavior and the intentions order in which they occur, the most relevant cluster is selected to fulfill the requester's present goals and needs. According to Algorithm 1, the similarity of current behavior and clustered behavior will be determined. As stated in section 3.1.1 although

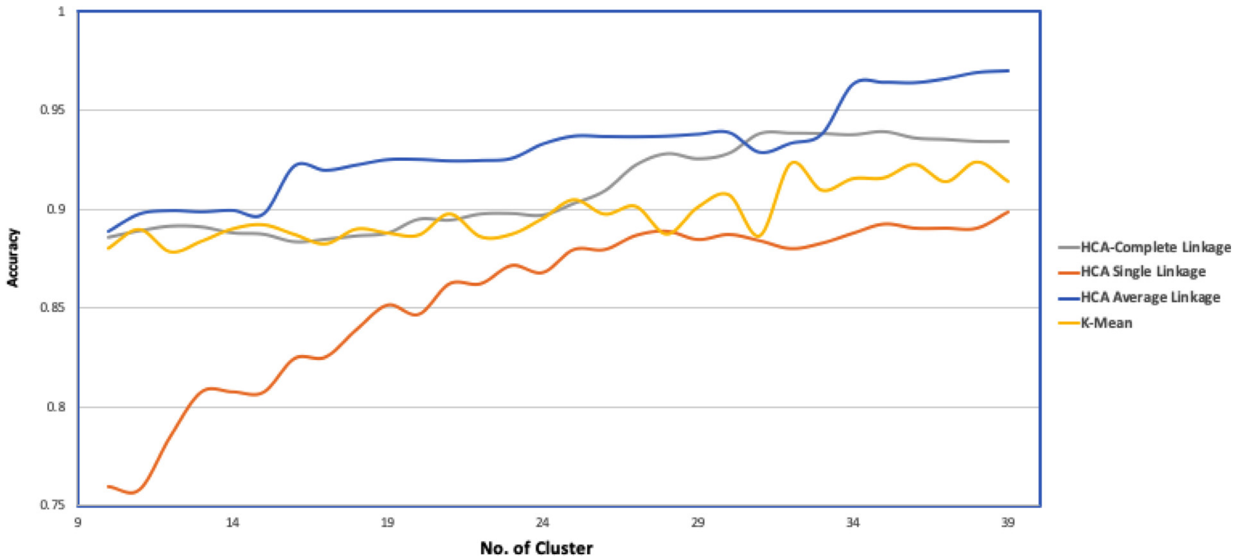

Fig. 3. Clustering Accuracy Evaluation.

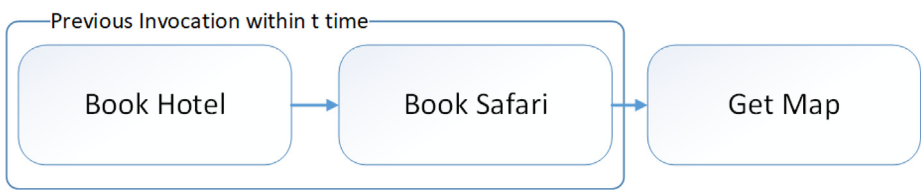

Fig. 4. GetMap Request for Entertainment Purposes.

having the same intents, a similar sequence of intention accomplishment may signify different behavior. Whereas the users who have similar historical records on some services would have similar behavior toward other services (Luo, 2015). Fig. 4 shows a user's behavior which illustrates the getMap service being invoked after the BookHotel and BookSafari services have been called within period of time. On the other hand, Fig. 5 indicates the same request for the getMap service after the FindEmergencyHospital and ReserveOperation-Room services have been invoked within period of time. As a result, the user's preference may evolve toward different behavior. The user may prioritize cost above speed in the first request, but in the second request, the user may swap those priorities.

3.2.2. QoS importance calculator

Users are more likely to share similar preferences when their circumstances are comparable, as previously indicated. The QoS importance for service selection is determined in this stage by estimating the QoS importance of a related cluster. Weights for each QoS are determined using the entropy approach. Measurement of a random variability's uncertainty is the goal of the entropy theory (Shannon, 2001). The more varied the data, the greater the degree of ambiguity, and the higher the significance of the criterion. The information on each decision can be expressed as an entropy value, and the relative importance of each criterion can be assessed in an objective manner (Liu, 2020). However, as the objective weights do not accurately reflect the user's preferences, we try to derive subjective weights from user behavior by exploiting the entropy of user's behavior history. The user's degree of satisfaction is reflected by how frequently they behave toward using the same service to attain specified goals in a given context. In this research, the entropy approach is used to record user preference based on the user's past behavior.

3.2.3. QoS criteria selector

In service selection, various domains utilize varying numbers of QoS criteria when choosing services. I it is known fact that as the number of QoS criteria rises, user-centric service selection becomes more challenging. For instance, the authors of (Baranwal et al., 2020) have identified 36 Quality of Service (QoS) metrics for IoT services that are grouped into three categories: computation, communication, and objects. The paper (Muñoz Frutos, 2009) describes 30 QoS in their suggested QoS model, which is subdivided into a number of categories. The effectiveness of the service selection mode is adversely impacted by considering all QoS. In the proposed model QoS criteria that are less important than the specified threshold will be deemed ineffective criteria and eliminated from the service selection process. For this reason, the threshold will be determined based on the standard deviation of weights, as shown in Eq. (4), and any criterion that has a weight less than its threshold has a negligible impact on the final selection.

$$k = \sum_{i=1}^{n} iifW^{i} \geq \sqrt[2]{\frac{\sum_{i=1}^{n}(W^{i} - \bar{W})^{2}}{n}} \tag{4}$$

Where k is the number of considered criteria, n is the total number of criteria, Wi is the weight of criteria i.

3.3. Selection

In this stage the candidate services will be selected based on their functionality matching with user's request. Subsequently the candidate services are ranked and the best service will be cho-

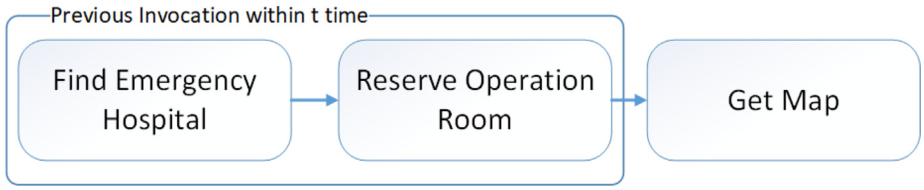

Fig. 5. GetMap Request for Emergency Purposes.

sen to carry out the user's objective. Most researchers utilized the optimization approach to reduce search space because there are many providers that offer the same functionality for a given user query. Popular optimization algorithms such as Skyline and p-dominant algorithms have been applied to optimize search space for service selection (Purohit and Kumar, 2019; Kertiou, 2018; Serrai, 2017). However, the number of QoS criteria has a significant impact on the performance of these algorithms, while high dimensionality leads to poor performance and efficiency, and vice versa (Peng and Chen, 2019). Therefore, the k-dominant algorithm has been utilized to optimize the search space, in which only the k QoS criterion with the highest priority will participate in the selection process. Rather than being superior in all dimensions, a point just has to be superior in k dimensions to dominate another point. Point A k-dominates point B if it's better than or equal to point B and its better in at least one of k dimensions (Papadopoulos, 2020). Compared to computing the skyline, computing the k-dominant is more challenging since it requires examining every possible dominance relationship between every pair of points (Peng and Chen, 2019). We introduce Dynamic K-Skyline, which selects the k relevant dimensions, as an alternative to considering each and every potential combination of k in an effort to find the optimal one. While the K relevant dimensions will be selected based on the importance of QoS and its effectiveness on service selection as described in section 3.2.3. The dynamic K-Skyline is presented in Algorithm 2 as in Fig. 6. Finally, the dominating service will be scored using the TOPSIS algorithm to return the top-ranked services to the customer.

The algorithm obtains the candidate services and the weights of the QoS, as shown in the algorithm. The QoS weights are calculated using the entropy weight method and user behavior. The line numbers 3 calculate the desired QoS attributes according to the equation to be used to optimize the search space. Lines 7–23 traverse all candidate services and add non-dominated services to the Skyline service list based on the selected k attributes from line 3.

4. Case study

In this section, we present a case study for the proposed model through a simple scenario in which a user desires to select a service to satisfy the "locate town" intention in a certain context. However, a user has already accomplished two additional intents during the previous 30 min, which are "get list of hospitals" and "rent car" intentions, respectively, as shown in Fig. 7. Hence, the user behavior is to select travel and location service after the invocation of healthcare service. The selection of services to satisfy the locate town intention will be based on the services listed in Table 1 which are location and map services derived from a real-world dataset, QWS (Al-Masri and Mahmoud, 2008).

Values in Table 1 are normalized using the min/max method, with 1 being the greatest value and most preferred for all criteria and 0 being the lowest value and least preferred. The QWS has 2507 services from different domains that measures nine different metrics of QoS: Response Time (RE), Availability (AV), Throughput (TH), Successability (SU), Reliability (RE), Compliance (CO), Best Practices (BP), Latency (LA), and Documentation (DO). In this study only four QoS metrics are considered for simplicity and illustration, which are RT, AV, TH, and RE. A comparison will be made between the user's behavior and the behavior history records to identify similar cases. For this reason, a historical user behavior dataset (Najar et al., 2015) is also employed. This dataset was derived from the well-known OWLS-TC dataset, which has been expanded to include Intentional and Contextual data in study (Najar et al., 2015). The only domain for which historical data was generated was the travel domain. To verify our model, the dataset was enlarged to encompass user intent and context for additional domains, such as healthcare and education. They also didn't consider the QoS values in their dataset, therefore we appended QoS values to each record based on the distribution of QoS values in the QWS dataset. This case study shows how the model can predict user preferences based on how the user has behaved in the past.

4.1. Cluster user behavior

The user behavior history dataset contains 500 unique user behaviors. This dataset captures the user behavior of ten distinct individuals in a variety of scenarios for three different domains. This number of records was chosen since it is sufficiently diverse to evaluate the performance of the clustering algorithm for the selected domains and intentions.

The contexts were primarily separated into two sorts, namely device context and human context, with distinct descriptions for each type. The user's behavior over a period of time includes at least one or multiple service invocations to accomplish certain goals. In this case study, a period of 30 min was chosen to associate present user intention with prior intentions for one behavior, in other words, any service invocation that occurred during 30 min was considered to be one user behavior. Using the similarity approach and clustering algorithm mentioned in Sections 3.1.1 and 3.1.2, the dataset was clustered. Fig. 8 depicts the clustering outcome, in which the user behavior was grouped into 39 distinct clusters.

4.2. Calculate QoS importance

As described in Sections 3.2.1 and 3.2.2, a user's query is captured and the importance of QoS is determined through finding the historical record of similar behavior and context of the user. In the case study, the user attempts to locate a town by finding services related to the town locating functionality, however, the user accomplishes two other intentions which are getting a list of hospitals and renting car intentions in a period of 30 min respectively as shown in Fig. 5. The query was made in the context of a human user. The chosen human profile contains the location, time, and age of the user. Thus, the user attempts to request service after invoking getListOfHospital, there is a high probability that the user needs to locate the town for achieving the same goal.

Consequently, in this stage, the cluster with the highest similarity in terms of user behavior and user context is selected. The historical QoS values of the selected cluster are returned as shown in Table 2, which is a representation of the QoS values of users who

Algorithm 2. Dynamic K-Skyline Algorithm

```
Input CandidateServics, Weights
Output SkylineServices
Function getSkylineServicesP(CandidateServics,Weights)
Begin
    k=getDynamicK(Weights) //According to Equation 4
    SkylineServices.add(CandidateServics(0))
    isDominated=False
    For all service in CandidateServics
    Begin
        For all skyService in SkylineServices
        Begin
            If(service(k) dominants skyService(k))
            Begin
                SkylineServices.remove(skyService)
            End
            Else If(skyService(k) dominates service(k))
            Begin
                isDominated=True
                Break
            End
        End
        If(isDominated is False)
        Begin
            SkylineServices.add(service)
        End
    End
    Return SkylineServices
End
```

Fig. 6. Dynamic K-Skyline Algorithm.

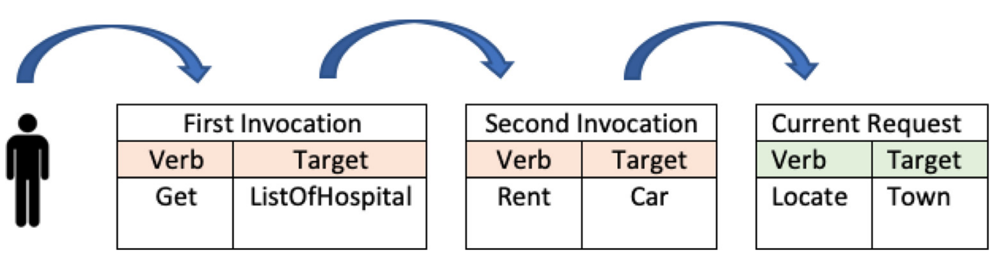

Fig. 7. Service Invocation Intention within Period of Time (30 min).

Table 1
Location Services Data in QWS.

ID	Service	RT	AV	TH	RE	Is Skyline
S1	basicmap	0.06	0.97	0.03	0.90	Yes
S2	MapService	0.09	0.39	0.14	0.82	No
S3	getMapwsdl	0.09	0.46	0.13	0.82	No
S4	remapService	0.14	0.85	0.32	0.75	No
S5	tmapService	0.18	0.94	0.06	0.65	No
S6	GetBestMapDefinitionService	0.25	0.93	0.14	0.75	No
S7	GenerateMapService	0.26	0.90	0.04	0.75	No
S8	GeocodeService	0.25	0.97	0.22	0.67	No
S9	CogoService	0.47	1.00	0.82	0.75	Yes
S10	DensityMap	0.50	0.92	0.19	0.82	Yes
S11	GeneidmapService	0.03	0.56	0.09	0.93	Yes
S12	CityLocations	0.37	0.86	0.52	0.82	Yes

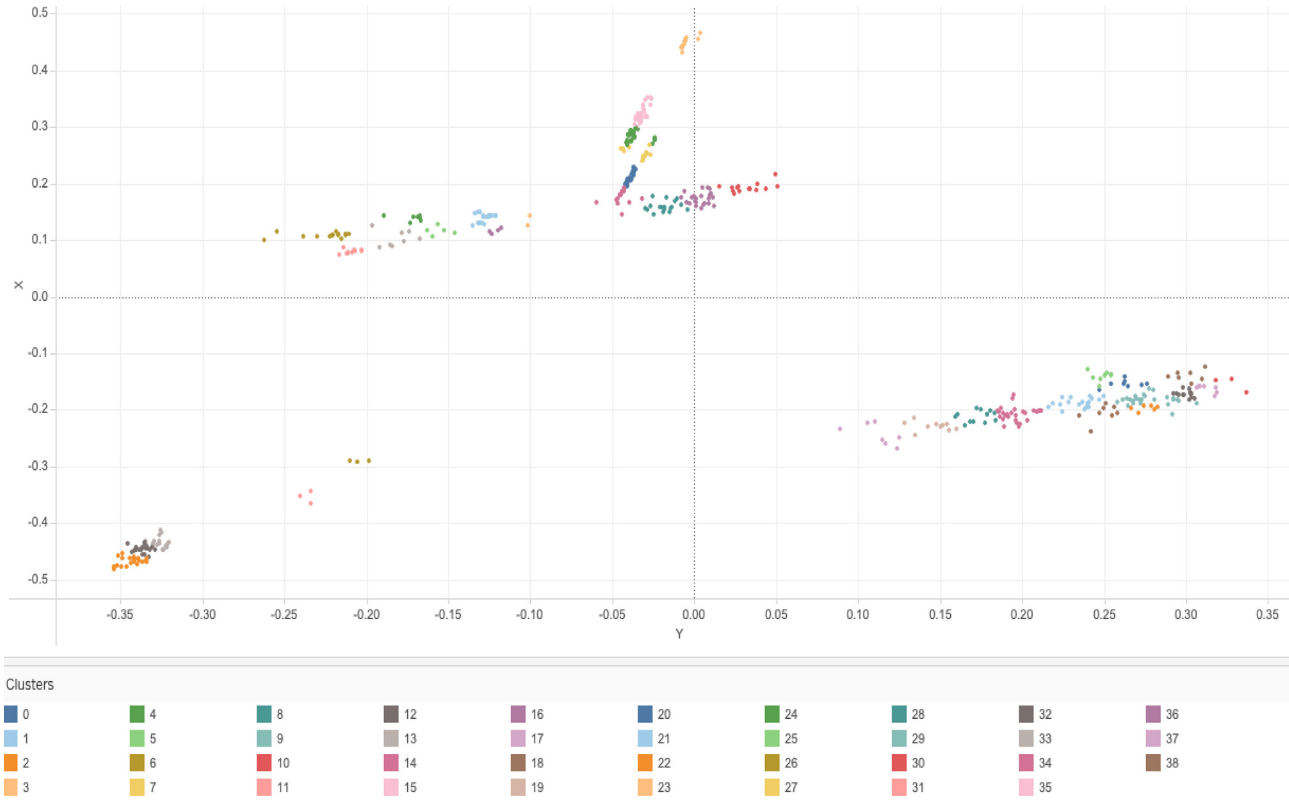

Fig. 8. Clustered User Behavior and Context.

had behavior and context comparable to that of the current user. According to Table 2, the returned cluster includes three users' behaviors that are similar to the current user query in terms of the number of intentions per behavior, the user context, and the combination of intentions that call for travel and health care services to achieve the same purpose.

The current user preference for locating town intention is calculated based on the historical user preferences. As stated in Section 3.2.2, after finding historical record for similar cluster the entropy weight is employed to determine the importance of QoS. The importance weight of each criterion is generated as in Table 3. The result shows the importance of response time over other QoS criteria regarding the user query. Furthermore, the effectiveness of QoS criteria was calculated according to Eq. (4), where only RT and AV are considered effective QoS criteria, therefore the other criteria are not going to be considered for selecting a service.

Table 3
Calculated Entropy Weight.

RT	AV	TH	RE
0.5	0.19	0.15	0.16

4.3. Service selection

The user expressed his/her intention to locate town after getting a list of hospital and renting a car respectively, therefore based on that intention and the proposed model the user preference predicted for that intention and behavior in a specific context without his/her intervention. Thus, using Algorithm 2 the user preference utilized to select desired service among available services listed in Table 1. After examining the weights of each QoS criterion, the dynamic K-Skyline algorithm utilizes effective and helpful

Table 2
Historical Record Result.

Intention ID	Domain	RT	AV	TH	RE
84	travel	251	81	53	64
84	healthcare	67	45	63	50
84	healthcare	58	57	60	65
105	healthcare	250	73	46	69
105	travel	58	77	74	49
105	travel	97	71	80	55
344	healthcare	224	37	30	65
344	healthcare	231	84	55	60
344	travel	114	95	49	61

criterion in accordance with Eq. (4). As stated in the previous section, only response time and availability are considered as effective criteria in this case study, because the historical behavior for that particular intention and context demonstrate the value of those criteria. Consequently, the services S9 and S10 are chosen as skyline services because they don't dominate each other in terms of response time and availability criteria. However, when the same user query is evaluated using all QoS parameters, five services are selected as skyline services as shown in Table 1. Finally, the TOPSIS is applied to rank skyline services based on estimated weights in Table 3. While S9 is chosen as the desirable service for the user regardless of choosing skyline or dynamic K-Skyline algorithms, S9 provides a supremum value of response time and availability as they are required by generated user preference.

Finally, this case study demonstrates the ability of the proposed model in prediction of user preference based on user intention and context without user intervention, also the effectiveness of QoS services are determined which consequence in reducing the size of skyline services in the selection stage of services.

5. Experiment

The presented model ability of predicting user preference and service selection are stated in the case study, however the model is evaluated to demonstrate the efficiency and robustness in selecting services based on predicted user preference. Thus, this section offers the experimental evaluation of the model in terms of the efficiency by determining the average execution time required to perform service selection on a dataset containing QWS. As aforementioned the proposed model use dynamic K-Skyline algorithm and TOPSIS for the selection, thus the proposed model compared to other model using full skyline algorithm and TOPSIS which is applied by many researchers for service selection. All experiments are carried out on a computer with a 2.90 GHz Intel Core i7 processor and 8 GB of RAM. Moreover, Java is the programming language used to code the algorithms.

5.1. Performance evaluation

To evaluate the performance of the proposed model we assume that all services in QWS dataset are offering the same functionality. In this step, we compare the proposed model to others in terms of the size of skyline services and the execution time of service selection. Nine arbitrary queries were selected for this purpose and executed for different QoS dimensions as in the dataset to indicate the size of the skyline. For each metric of QoS, each of these nine queries was run ten times. After that, the average execution time of the suggested model was calculated to evaluate its performance. Following that, the findings were evaluated in light of the full-skyline algorithm, which is a method that has been applied in a significant number of studies (Serrai, 2017; Liang et al., 2019; Barge et al., 2021), and Absolute Dominance (Jin, 2016; Jin, 2014). The result of the size of skyline services are shown in Fig. 9 which indicate the size of skyline service for various QoS dimensions. The outcome depicts the impact of considered the quantity QoS criteria on the size of the skyline. The size of skyline services decreases as fewer QoS dimensions are considered, given that the proposed method dynamically disregards inefficient criteria. Consequently, fewer indicators are examined for service selection, resulting in a reduction in the amount of skyline services. This proves that the proposed model facilitates an efficient service selection. On the other hand, the full-skyline considers all QoS criteria for service selection, which results in the skyline having a larger size. In the case of the absolute dominance method, a service A is considered to be dominant over a service B in the case that the minimum (QoS) value of A is higher than the maximum (QoS) value of B. This method operates well with a limited number of QoS dimensions. The result demonstrates that the method is inefficient with an increasing number of QoS dimensions. In comparison to other algorithms, it will provide the larger size of skyline services.

AS the number of Skyline services increase, the efficiency and performance of the proposed selection model will also decrease. To support this claim, the model has been tested on different queries and each query has been executed multiple times considering various quantities of QoS metrics. Fig. 10 shows the execution time for three models, it is clear that the proposed model results in improved performance. As a consequence of removing useless criteria, the suggested model performs better as the number of dimensions increases compared to other models. In order to demonstrate the impact of the number of examined QoS features on execution time reduction, the suggested algorithm is implemented using Block Nested Loop (BNL) since the benchmark studies use the same technique. Hence, the time complexity of all three techniques is $O(n^2 * p)$, where n represents the number of services and p represents the number of p. Nonetheless, the reaction time of

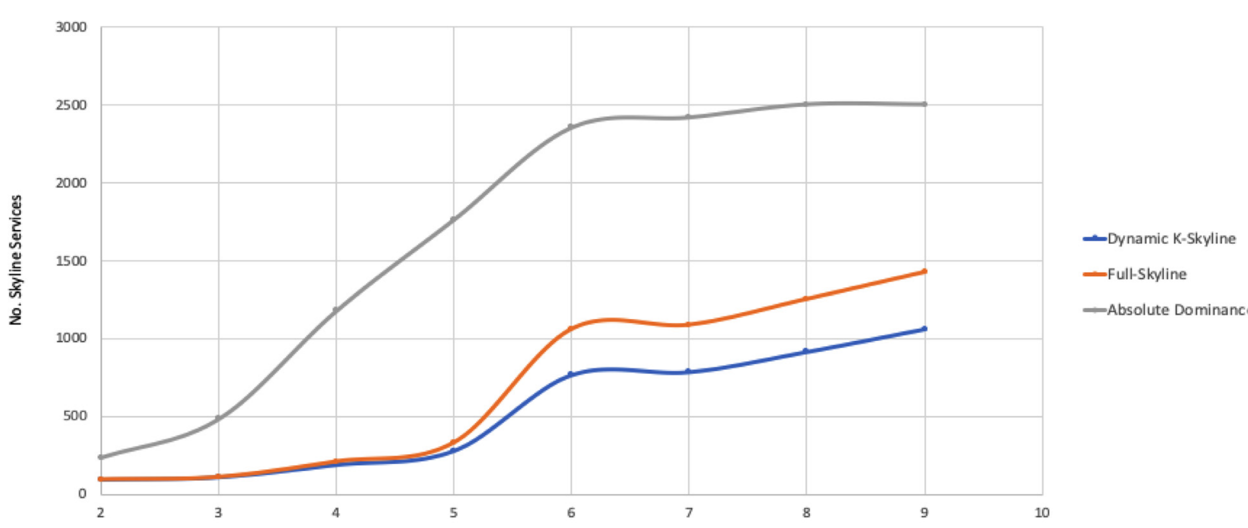

Fig. 9. Size of Skyline Services with Varied QoS Dimensions.

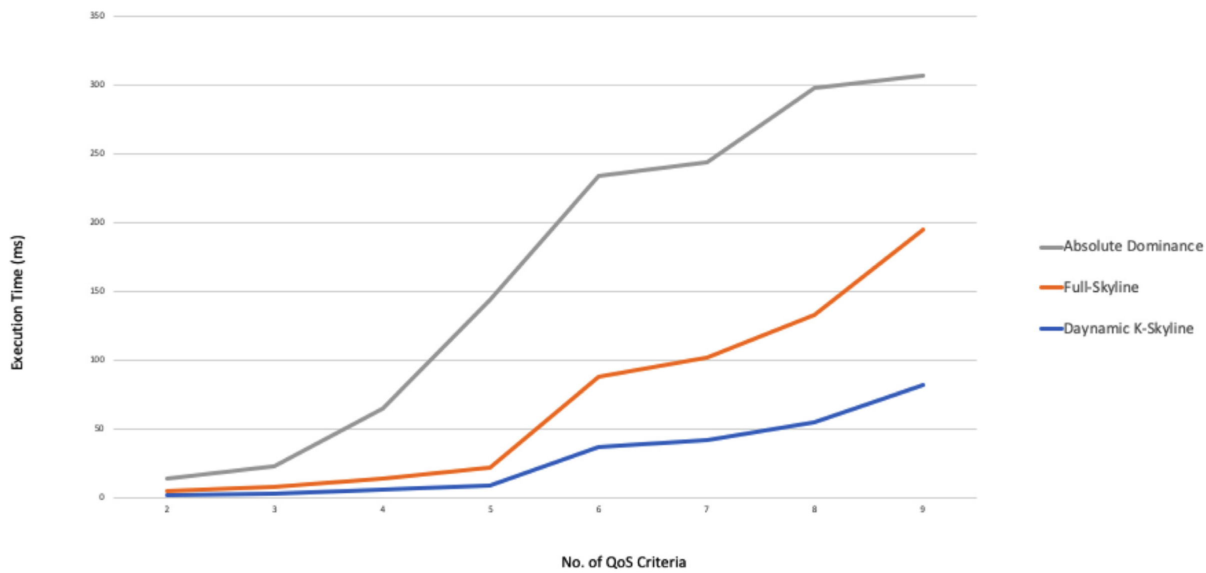

Fig. 10. The Average Response Time with Varied QoS Dimensions.

the suggested algorithm is relatively less than the other algorithms, as the number comparisons are conducted with fewer dimensions. Additionally, the performance gaps between the proposed model and alternative models grow as the number of dimensions increases. Moreover, the number of ineffective criteria varies based on user behavior history and user query, whereas in some cases, even with high dimensionality, the number of ineffective criteria is zero because, based on the user's behavior history, all the criteria are relatively important for service selection in that particular behavior.

5.2. Sensitivity analysis

The proposed model utilizes the user behavior and context to find user preference also the proportional importance of criteria for each service selection may differ from one user to the next based on their intention and context. As a result, the weight of a criterion may change if the user's intention and context change. If the order of ranking changes when the weights of the criteria change, the result is considered sensitive. The result is robust if the order of ranking doesn't change. Therefore, we assume that all QWS record provide same functionality and a random query based on existing user's behavior in dataset of behavior's history applied to calculate top services using dynamic K-Skyline and full-skyline algorithms. The TOPSIS are utilized for ranking skyline service in both algorithms. The importance of QoS are calculated based on user's query as shown in Table 4. The dynamic K-Skyline utilizes fewer dimensions than the skyline algorithm, and the neglected QoS considered as ineffective QoS criteria according to Eq. (4). We choose four dimensions as the bare minimum for QoS, since the likelihood of ineffective service decreases as the number of dimensions decreases. The top-ranked services are returned for each dimension for both algorithms.

In order to generate the ranking, the top 10 services are retrieved and ranked using both algorithms. Dynamic K-Skyline and TOPSIS are implemented after the full-skyline and TOPSIS algorithms have been applied. As shown in the previous section, the proposed model outperformed other models in terms of efficiency. In this section, the result of the ranking was evaluated for its robustness. For that purpose, sensitivity analysis is applied, which can determine the robustness. The result is considered robust if the top services for both models are similar. On the other hand, the result is considered sensitive to dissimilar top services. The comparison result between the proposed model and full-skyline is depicted in Fig. 11, which displays the ranking of the top 10 skyline services.

From the presented results, we can observe that the top ten skyline services for both models have high similarity in all QoS dimensions scenarios, with slight changes in the ranking of alternatives. This indicates the robustness of the presented model while the selected top services are almost the same even if some of the criteria are not used in the selection process. The top 1, 5, 6, and 7 services are the same for both models even with neglected ineffective QoS criteria in all scenarios. In contrast, the top 2, 3, 4, 8, 9, and 10 services are highly similar in both models with a slight change in alternatives. For instance, the top 2 and 3 services are swapped when we include nine QoS dimensions in dynamic K-Skyline algorithms, due to variations in QoS importance dependent on user intention in a certain context.

Table 4
QoS Importance based on Intention and Context.

	C1	C2	C3	C4	C5	C6	C7	C8	C9
QoS4	0.0921	0.3357	0.3744	0.1978					
QoS5	0.0647	0.2360	0.2633	0.1390	0.2969				
QoS6	0.0569	0.2073	0.2313	0.1221	0.2608	0.1216			
QoS7	0.0437	0.1595	0.1779	0.0940	0.2007	0.0935	0.2306		
QoS8	0.0421	0.1536	0.1714	0.0905	0.1933	0.0901	0.2221	0.0369	
QoS9	0.0396	0.1445	0.1611	0.0851	0.1817	0.0847	0.2088	0.0347	0.0597

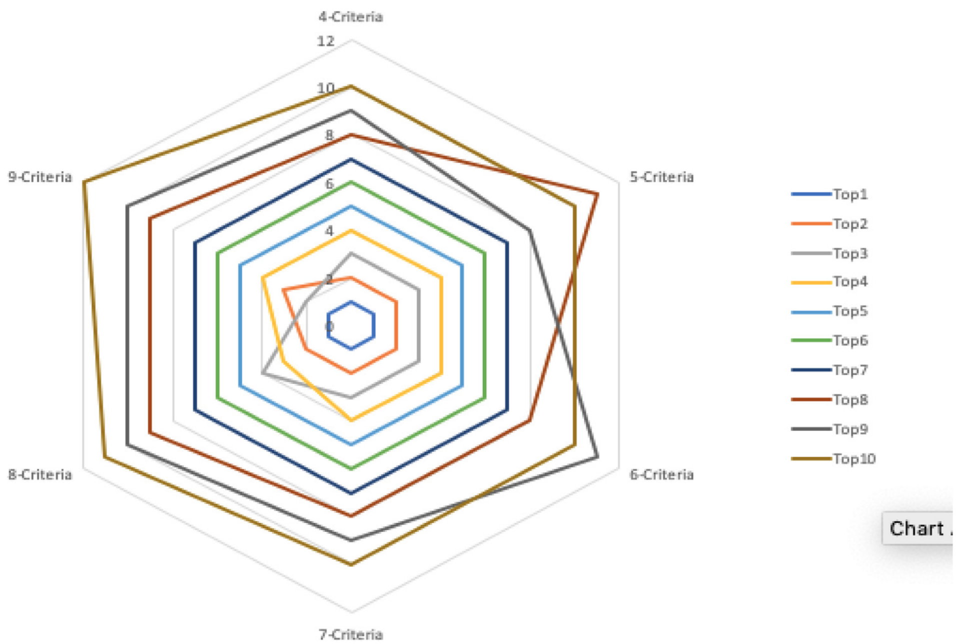

Fig. 11. Ranking Top 10 Skyline Services.

6. Discussion

In the rapidly evolving internet and IoT market, service selection is an important issue as it directly effects the experience of users. To understand the implementation of an application and various services such as IoT services, QoS criteria can be a good asset and requirements. QoS metrics may vary from one user to another and also may change based on user behavior and context, therefore, defining QoS criteria for each application and behavior is a challenging task. The significance of QoS criteria, as well as the influences of user intention and context on service selection are examined in this paper. The user must determine the significance of QoS based on user desire in user-centric service selection. User intention and context must be taken into consideration when determining user preference because the user's preferences change as their context and intentions change. This study identified a research gap in defining user preference based on user intention and context, as there are a few studies that took user intention and context into account to predict user selection. In order to select the services that the user needs the most, this research has presented a new service selection model that uses user intention and context to learn about user behavior and preference.

The primary contribution of this study is the model suggested, which employs a number of components and methods. Three distinct components comprise the model: clustering, pre-selection, and selection. The clustering component aggregates user behaviors discovered over time based on user intents and context, allowing comparable behaviors to be discovered for future users' intentions in a particular context.

The pre-selection and selection components are responsible for handling user inquiries. Pre-selection records user behavior and context, then compares it to the existing cluster to identify the behavior and context that are most comparable. Using the entropy weighting method, the history of similar behavior is retrieved to evaluate the importance of QoS criterion. In addition, in pre-selection, the effectiveness of the QoS is calculated to minimize the dimensions of the QoS.

The model's selection component gathers user queries and uses the QoS weights for effective criteria from the pre-selection component. This element shows the dynamic K-Skyline algorithm with TOPSIS for optimizing the search space among available services and eventually ranking the top services for selection.

In order to evaluate the ability of the presented model, a case study is presented, demonstrating how the model predicts a user's preference dynamically without user involvement based on the user's behavior and context. The case study was presented based on the real-world dataset for available services and a dataset for users' behavior history. As the case study shows the ability of the proposed model, the effectiveness and robustness of the provided model are evaluated through experiments. Experiments were also performed on a real dataset.

The efficiency of the proposed model is compared to other models, and the proposed model outperforms the existing models since the proposed model determines the QoS criteria from the user behavior and the ineffective QoS are neglected in the selection process, which leads to retrieving a smaller size of optimal services and eventually a smaller execution time. The result is required to show the robustness, so the sensitivity analysis performed shows the robustness of the proposed model. The results show that the model can accurately predict user preferences based on similar behavior histories and can choose the best service for the user based on their intentions, context, and behavior.

7. Conclusion and future direction

This study has identified a major problem with the current service selection and preference of the user. The proposed model has emphasized the user behavior, and context to predict user preference for service selection. For that purpose, the history of user intentions and behaviors in a certain context are considered to generate user preference. This uncovered a new challenge which is the size and sparsity of historical data, to address the challenge, the proposed model utilizes a clustering approach to group similar historical behavior and context. Thus, the hierarchical clustering algorithm-average linkage is chosen after evaluating it in relation to another clustering algorithms. The proposed model evaluates the user's preference based on similar past behavior and context, as well as the QoS's degree of effectiveness, to differentiate them for service selection. The proposed model optimizes the search space based on effective QoS metrics using proposed dynamic K-Skyline algorithm to optimize search space and select skyline services in an efficient way. Finally, the TOPSIS algorithm is presented to rank alternatives and return top services for selection purposes. This model helps service requesters to find their dynamic preferences based on different intentions and contexts without any human interaction or expert preferences.

The proposed model is evaluated for its efficiency and robustness in service selection on a real-world dataset. The evaluation is done by calculating the size of the skyline and the response time of a returning skyline service, the result strongly supports the efficiency of the presented model. Moreover, the robustness of the model is assessed through a sensitivity analysis of the final service ranking, and the result supports the model's robustness.

This research has some limitations. One of the limitations is the lack of a public dataset for user behavior in service selection that considers QoS services across many domains and industries. There is only one dataset available, which is restricted to historical users' intentions for picking travel domain services without taking QoS into account. The other datasets are collected to capture QoS service by evaluating the available services, considering variances in geographical location, but without capturing actual user intent. As another limitation, the study mainly considered median thresholds for clustering and dynamic QoS attribute selection instead of dynamic thresholds, where varying the threshold affects the precision of clustering behavior.

When a group of comparable behaviors and situations has utilized the same service multiple times for similar goals, it is possible to predict the users' preferences, which means that a new user behavior employs equal weights for the QoS. The efficiency of the suggested service selection algorithm may also change depending on the threshold used to choose the k most important features. Because user behavior is composed of multiple intentions and contexts that might occur in various orders, the data has a significant degree of overlap, reducing the accuracy of the clustering algorithm.

The future direction of this study is to further improve the model by enhancing the clustering algorithm to achieve better accuracy in clustering user behavior and context. In addition, the weighting mechanism to find the weight of the historical behavior and context requires an enhancement to distinguish the negative and positive changes in user preferences toward a specific QoS criteria in user behavior. One more direction of this study is to consider service composition for each intention while the current study has considered each user intention in their behavior as atomic service invocation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We partially acknowledge Universiti Teknologi Malaysia UTMHR Grant Vot No. 08G67 which has made this research endeavor possible.

References

Abosaif, A.N., Hamza, H.S., 2020. Quality of service-aware service selection algorithms for the internet of things environment: a review paper. Array 8, 100041.
Alaoui, S., Idrissi, Y.E.B.E., Ajhoun, R., 2015. Building rich user profile based on intentional perspective. Procedia Comput. Sci. 73, 342–349.
Ali, S., Kibria, M.G., Chong, I., 2017. WoO enabled IoT service provisioning based on learning user preferences and situation. In: 2017 International Conference on Information Networking (ICOIN). IEEE.
Al-Masri, E., Mahmoud, Q.H., 2008. Investigating web services on the world wide web. In: Proceedings of the 17th international conference on World Wide Web. ACM.
Baranwal, G., Singh, M., Vidyarthi, D.P., 2020. A framework for IoT service selection. J. Supercomput. 76 (4), 2777–2814.
Barge, Y., Purohit, L., Saha, S., 2021. A Skyline Based Technique for Web Service Selection. In: Smart Computing Techniques and Applications. Springer, pp. 461–471.
Benouaret, K., Benslimane, D., Hadjali, A., 2012. Selecting skyline web services from uncertain QoS. Proceedings - 2012 IEEE 9th International Conference on Services Computing, SCC 2012, pp. 523-530.
Cong, Z. et al., 2015. Service discovery acceleration with hierarchical clustering. Inf. Syst. Front. 17 (4), 799–808.
Daosabah, A., Guermah, H., Nassar, M., 2019. Dynamic composition of services: An overview of approaches led by the context and intent of the user. In: Proceedings of the 4th International Conference on Big Data and Internet of Things.
Daosabah, A., et al., 2021. Integrating Context and Intention for Optimal Semantic Web Service Composition Using AI Planning. In: 2021 4th International Conference on Advanced Communication Technologies and Networking (CommNet). IEEE.
Eyal Salman, H. et al., 2018. Semantic clustering of functional requirements using agglomerative hierarchical clustering. Information 9 (9), 222.
Gochhayat, S.P. et al., 2019. LISA: Lightweight context-aware IoT service architecture. J. Clean. Prod. 212, 1345–1356.
Jaafar, A.A. et al., 2019. Internet of Thing and Smart City: State of the Art and Future Trends. In: Advances in Computer Communication and Computational Sciences. Springer, pp. 3–28.
Jin, X. et al., 2014. IoT service selection based on physical service model and absolute dominance relationship. in 2014 IEEE 7th International Conference on Service-Oriented Computing and Applications. IEEE.
Jin, X. et al., 2016. A fast and scalable approach for IoT service selection based on a physical service model. Inf. Syst. Front.
Kanagaraju, P., Nallusamy, R., 2018. Registry service selection based secured Internet of Things with imperative control for industrial applications. Clust. Comput., 1–13
Kaur, G., Tomar, P., Singh, P., 2018. Design of cloud-based green IoT architecture for smart cities. Internet of Things and Big Data Analytics Toward Next-Generation Intelligence, 315–333.
Kertiou, I. et al., 2018. A dynamic skyline technique for a context-aware selection of the best sensors in an IoT architecture. Ad Hoc Netw. 81, 183–196.
Liang, X., Lu, Q., Li. M., 2019. Research on web service selection based on improved skyline algorithm. in 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). IEEE.
Liu, M. et al., 2020. A heterogeneous QoS-based cloud service selection approach using entropy weight and GRA-ELECTRE III. Math. Probl. Eng. 2020.
Liu, Y., Ngu, A.H., Zeng, L.Z., 2004. QoS computation and policing in dynamic web service selection. Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters - WWW Alt. '04, p. 66.
Luo, X. et al., 2015. Web Service QoS Prediction Based on Adaptive Dynamic Programming Using Fuzzy Neural Networks for Cloud Services. IEEE Access 3, 2260–2269.
Muñoz Frutos, H. et al., 2009. Enhancing service selection by semantic qos. European Semantic Web Conference. Springer.
Najar, S., Kirsch-Pinheiro, M., Souveyet, C., 2011. The influence of context on intentional service. In: 2011 IEEE 35th Annual Computer Software and Applications Conference Workshops. IEEE.
Najar, S., Pinheiro, M.K., Souveyet, C., 2014. A new approach for service discovery and prediction on pervasive information system. Procedia Comput. Sci. 32, 421–428.
Najar, S., Pinheiro, M.K., Souveyet, C., 2015. Service discovery and prediction on Pervasive Information System. J. Ambient Intell. Hum. Comput. 6 (4), 407–423.
Natarajan, B. et al., 2020. New clustering-based semantic service selection and user preferential model. IEEE Syst. J.
Nizamkari, N.S., 2017. A graph-based trust-enhanced recommender system for service selection in IOT. In: Inventive Systems and Control (ICISC), 2017 International Conference on. 2017. IEEE.
Nwe, N.H.W., Bao, J.M., Cui, G., 2014. Flexible user-centric service selection algorithm for Internet of Things services. J. China Univ. Posts Telecommun. 21, 64–70.
Papadopoulos, A.N. et al., 2020. Skylines and other dominance-based queries. Synthesis Lectures on Data Manage. 15 (2), 1–158.
Peng, Y.-W., Chen, W.-M., 2019. Parallel k-dominant skyline queries in high-dimensional datasets. Inf. Sci. 496, 538–552.
Purohit, L., Kumar, S., 2019. Clustering based approach for web service selection using skyline computations. In: 2019 IEEE international conference on web services (ICWS). IEEE.
Qi, L. et al., 2017. "Time–Location–Frequency"–aware Internet of things service selection based on historical records. Int. J. Distributed Sensor Netw. 13 (1). 1550147716688696.

Qi, L., et al., 2015. A context-aware service selection approach based on historical records. In: Cloud Computing and Big Data (CCBD), 2015 International Conference on. 2015. IEEE.

Rousseeuw, P.J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. 20, 53–65.

Salim, T.A. et al., 2020. Effects of smart city service channel-and user-characteristics on user satisfaction and continuance intention. Inf. Technol. People.

Serrai, W. et al., 2017. Towards an efficient and a more accurate web service selection using MCDM methods. J. Computational Sci. 22, 253–267.

Shannon, C.E., 2001. A mathematical theory of communication. ACM SIGMOBILE mobile computing and communications review 5 (1), 3–55.

Singh, M., Baranwal, G., Tripathi, A.K., 2020. QoS-aware selection of IoT-based service. Arab. J. Sci. Eng. 45 (12), 10033–10050.

Wang, Y. et al., 2020. Effective user preference clustering in web service applications. Comput. J. 63 (11), 1633–1643.

Wei, Q. et al., 2016. Lightweight semantic service modelling for IoT: an environment-based approach. Int. J. Embed. Syst. 8 (2–3), 164–173.

Xiang, C. et al., 2015. QoS-based service selection with lightweight description for large-scale service-oriented internet of things. Tsinghua Sci. Technol. 20 (4), 336–347.

Xu, Y. et al., 2016. Context-aware QoS prediction for web service recommendation and selection. Expert Syst. Appl. 53, 75–86.

Yang, H.-L., Lin, S.-L., 2015. User continuance intention to use cloud storage service. Comput. Hum. Behav. 52, 219–232.

Yu, H.Q. and S. Rei, Non-functional property based service selection : A survey and classification of approaches.

Zhao, Y., et al., 2017. Automatically learning user preferences for personalized service composition. In 2017 IEEE International Conference on Web Services (ICWS). 201IEEE.