Knowledge-Based Decision Support System for Selecting Optimal Web Services Based on QoS Attributes for Business Process Composition

Stipan Podobnic[1,*], Barbara Charchekhandra[2]
[1]Department of Mathematics, University of Rijeka, City of Rijeka, Croatia
[2]Jadavpur University, Department of Mathematics, Kolkata, India
Emails: Stipanpod133@Uniri.hr; Charchekhandrabar32@yahoo.com

Abstract

Web services are a crucial part of large-scale software development and cross-organizational collaboration. This chapter discusses the challenges of selecting the finest internet services among the vast array of possibilities available, with an emphasis on quality of service (QoS) features. Web services must fulfil every requirement needed to provide optimal user experience and the efficient execution of corporate operations. In order to find the best services, we look at important quality of service characteristics including response speed, reliability, accessibility, and efficiency. In what follows, you will find a detailed method for selecting services. The approach consists of three steps: finding services, improving them according to QoS constraints, and grading those using weighted normalized techniques. At each stage, methods are provided to ensure an accurate and successful selection that meets the customer's needs. The proposed method seems to work, according to the results of the trials. The rating of services for several customers with varying limits, achieved using real-life data sets, demonstrates the approach of filtering and assessing to acquire optimal results. This method boosts the efficiency and usefulness of the selected services by combining functional and non-functional aspects. Finally, this part concludes by stressing the importance of quality of service in guaranteeing customer satisfaction and optimizing the delivery of services in competitive and fast-changing environments. Service 3 has the highest accuracy rate at 96.5%. Due to their low reaction times and high availability, Services 2 and 6 are in close second place. Services 4 and 7 have good availability ratings; however, they take longer to respond. Services 1 and 8 have moderate availability and high response times; hence, they get the lowest scores. When it comes to reliability and accuracy, Service 3 remains your most effective choice.

Keywords: B2C; B2B; G2B; QoS; WSDL

1. Introduction

Web services have seen an explosion in popularity, which has resulted in a transformation in the way that businesses produce and distribute solutions for software [1-2]. When it comes to contemporary computing, web services are a vital tool for the development of large-scale, distributed programs. They do this by combining the functions that are offered by a variety of service providers. In light of the fact that businesses are becoming more and more dependent on such offerings to support business-to-business (B2B), business-to-consumer (B2C), government-to-business (G2B), as well as various types of digital interactions, it is becoming more important to pick the most appropriate web services. One of the most important challenges in service-oriented computation is addressed in the previous section, which focuses on the development of an effective approach for choosing crucial web services based on QoS qualities.

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

The internet service choosing is not only an exploration procedure; rather, it requires determining which offering is best appropriate in terms of meeting both non-functional as well as functional needs. For example, the tasks, which the provider of the service is responsible for, are considered practical demands, while non-functional criteria include things like reliability, efficiency, and scalability [3]. Conventional methods of service identification often fail to meet the aspirations of users since they are unable to assess services beyond the functions that are offered for them. The only way to address this gap is to use methods that focus on service quality. In order to ensure that the chosen service meets its financial obligations and enhances customer happiness, these strategies optimize critical parameters including response time, effectiveness, and cost [4-5]. It is impossible to overstate the significance of quality of service when selecting an online service. Companies have several constraints, including procedures that are time-sensitive, financial restraints, and varying degrees of reliability. For instance, in the world of stock platforms trading, substantial financial losses may occur if stock prices were to be delayed. Furthermore, patients may have major consequences because of subpar healthcare treatments. Incorporating considerations of service quality into the decision-making process ensures that the services precisely meet the user's needs within their specified constraints.

It is an orderly approach to service selection that forms the basis of the method described here [6-7]. Discovering services is the first step in the method. This entails determining which solutions are practical based on the functional requirements and requirements. Service refining follows, which entails thinning down options based on non-functional attributes. The penultimate step is service grading, which comprises ranking services by calculating normalized and weighted quality of service ratings. In order to set up rapid and precise service selection processes, an entire structure is given, with specific instructions for each stage [8-9]. An important aspect of this method is the use of QoS attributes. These factors are associated with four types of quality of service: security-related, configuration-and cost controlling, transaction-related, and runtime-related. Considering their direct impact on the user experience and operating success of the framework, runtime criteria such as response time, reliability, accessibility, and efficacy are prioritized among these. A number of measures are response time, which measures the speed that an organization responds to a request, reliability, which measures how well the service continues to function over time, and availability, which measures the likelihood of whether the service is available when demanded at any particular time. The proposed approach also makes use of new technologies that are intended to deal with specific challenges that arise while choosing internet services [10]. For the purpose of to find potential services, the Service Tracking Algorithms examine the Web Services Description Language (WSDL) files that contain details about the service's characteristics.

The Services Grade Algorithms is responsible for calculating final ratings according to a weighed normalizing of QOS criteria, while the Service Refined Algorithms is responsible for eliminating services that are not acceptable by applying user-defined limitations over them [11-12]. With this organized technique, the choice process is guaranteed to be both scalable as well as applicable to a wide range of application areas. In addition to this, the sections highlight the limits of the approaches that are now in use, such as computational biology, memetic techniques, and optimization using particle swarms. In spite of the fact that these methods provide answers to the issue regarding service choosing, they often demonstrate difficulties in terms of scalability, computing complexities, or rigidity when it comes to managing evolving user needs. These inadequacies are addressed by the technique that has been developed, which incorporates a constraint-driven methodology that is adaptable and responsive to changing service architectures.

During the course of the section, verification by experiment is an essential component. This approach is used to situations that include several customers with different quality of service limitations by making use of datasets taken from the actual world [13-14]. The services that are provided to each individual customer are monitored, improved, and rated, which ultimately results in a ranking of the best services. The findings demonstrate that the technique that was presented is successful in obtaining high levels of contentment while also striking a balance between the goals that were set by the users.

It improves their importance and quality of chosen services by combining both non-functional and functional aspects, so tackling significant difficulties that are faced by contemporary businesses. In addition to enhancing the effectiveness of service selection, the strategy that has been presented results in the establishment of an architecture that is both flexible and extensible to a wide variety of possible applications domain [15]. With this effort, the groundwork is laid for future developments in service-oriented structures, which will ensure that web-based applications will continue to satisfy the ever-increasing needs of a world that is more linked via digital means. Markets and to provide economic value for the organization. Products and services may also be considered innovations.

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

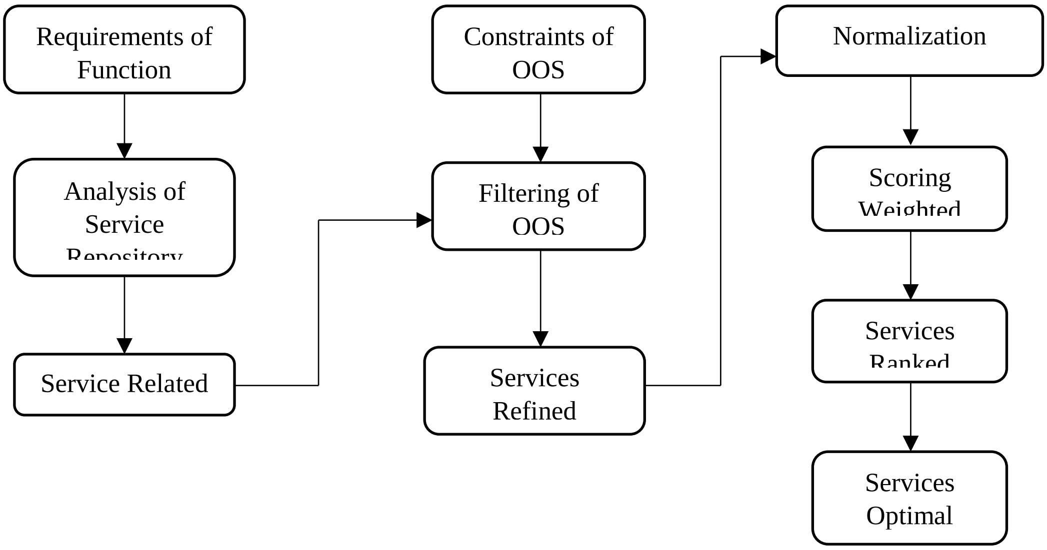

Figure 1. Visual representation of the steps used to choose a web service according to QoS criteria.

This block diagram shows the process for choosing the best web services according to QoS criteria. In the first step, known as discovering services, a procedure called service repositories evaluation is used to input the user's functional specifications [16]. At this point, the services from the repository are located, usually with the help of information found in WSDL files. A collection of interconnected services that fulfil the user-specified functional requirements is the result of this phase. At this point, we have made careful to choose a large pool of services to evaluate further.

Services refinement is the next phase; it uses the user-supplied QoS limitations to reduce the relevant services. Time to answer, dependability, along with accessibility are all examples of factors that may be considered limitations [17-18]. The QoS limitations and associated services discovered in the analysis stage make up the data that goes to this step. Quality of service filtration is applied to them, removing any services that do not fulfil the specified criteria. The refined services are the final product, and they comprise a selection of items that meet both functional and not-functional criteria.

Finally, we assess the services by normalized and weighing their ratings, which we do in relation to the enhanced services [19]. Following the standardization of QoS parameters in the normalization step, which guarantees continuity, the weighted scoring technique ranks and evaluates the services based on the user-set priorities. The final output is a ranked list of options, with the top service taking precedence. By picking an appropriate service with outstanding amenities and outcomes, this systematic approach enhances efficiency in operations and user delight.

2. Related Work

Researchers in the field of service-oriented computing have recently made internet provider selection a top research priority. In order to find and assess apps that cater to both practical and non-practical requirements, many methods have been proposed. The author offered as one of the first methods for prioritizing internet services based on QoS considerations [20]. This approach shifted the focus on the importance of user feedback and the precision of service track in live environments. As part of their strategy, they implemented feedback chains to adjust service ranks in response to real-time efficiency metrics. Despite their method's effectiveness in capturing user enjoyment, scaling proved challenging due to the tedious nature of continuous monitoring.

For service creation and selection that is QoS aware, the researcher proposed a method based on a quadratic programming. This approach included concurrently optimizing numerous QoS factors, such as speed of response, cost, and reliability [21-22]. When faced with situations that required decision-making based on many criteria, this strategy proved to be useful since it provided an organized framework for considering trade-offs. The computing expenses of solving linear algorithm models, on the other hand, hindered its application in real-time contexts that had huge-scale services repository. In a comparable way, the investigator expanded upon this concept by using adaptive algorithms to solve the issue with service selection. This was especially effective in dynamic environments where quality of service characteristics was expected to undergo changes. In spite of the

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

fact that genetic algorithms exhibited a high degree of adaptability and versatility, they often necessitated a substantial amount of computing power and the calibration of algorithms parameter in order to produce the best possible outcomes.

In their contribution to the area, the author proposed a skyline-based strategy for service choice. The objective of this technique was to determine which offerings were Pareto-optimal in terms of quality of their features as well as which solutions were not as good as others were. The field of search was greatly reduced by this strategy, which made it far more effective than approaches that covered every possible scenario [23-24]. The dependence on Pareto dominance, on the other hand, rendered the technique less successful in situations where there were numerous opposing QoS features. This was because the technique had difficulty striking a balance between the trade-offs among various services. The investigator, on the other hand, concentrated their efforts on a QoS-aware middleware programs that permitted dynamic association of internet services. The fact that they are the middleware technology offered real-time quality of service evaluation and service adaption made it exceptionally well-suited for situations that are prone to change. However, the dependence on middleware infrastructures resulted in the introduction of extra complexity and expense.

The investigator the application of utility theory to the selection of online services. He presented a model that allocated utility ratings to services depending on the QoS characteristics of such services. By using this technique, consumers were able to establish weights for each QoS characteristic, which enabled them to pick individualized services [25]. The adaptability of utility-based models made them suitable to a broad range of situations; nevertheless, they needed accurate input from users indicating their preferences, which was not always practicable in real-world circumstances. The researcher enhanced utility-based methodologies by including global optimization techniques in order to solve the challenge of service selection in composite systems Improving scalability was the end outcome of an arrangement among regional and international quality of service constraints. Upon the other side, each of the elements' efficiency could be subpar due to an over-reliance on overall optimization.

For how to choose a service, the author offered a particle swarm optimization (PSO) method. The limitations of conventional methods in handling large-scale problems inspired the creation of this program. In cases involving changing quality of service features, PSO's exceptional convergence skills and adaptability were shown. Despite these advantages, the results of the algorithm were quite sensitive to the variables used, which required domain expertise for its successful execution. The investigator used a method known as ant colony optimization (ACO) in order to solve the issue of service selection. This was accomplished by using the system's capacity to concurrently investigate numerous potential solutions. However, ACO had difficulties in terms of computing efficiency, especially when dealing with huge service repositories. Despite its success in identifying near-optimal solutions in complicated situations, ACO experienced hurdles.

In recent years, strategies that use machine learning have become more prevalent in the selection of online services. The investigator suggested a supervised learning framework. This framework was able to predict the QoS performance of services by using historical data. Through the process of training models on previous observations, their technique dramatically enhanced the accuracy of service selection processes. On the other hand, the fact that it relied on high-quality training data restricted its application in fields that had datasets that were either sparse or noisy. The author, who included deep learning methods in order to capture the intricate correlations that exist between QoS variables, expanded upon this concept. However, despite the fact that deep learning models had improved predictive skills, they used a significant amount of computing resources and presented difficulties in terms of interpretability.

Authors used ontologies in order to improve the accuracy of functional and non-functional matching. These authors investigated the incorporation of semantic approaches into the process of service selection. When it came to domains that had well defined ontologies, semantic techniques proved to be very beneficial since they made it easier to pick services that were aware of the context. Nevertheless, the development and upkeep of domain-specific ontologies brought to an increase in the level of complexity and the number of resources that were requested. In a similar vein, the researcher suggested a hybrid technique that combines semantic matching with QoS optimization. This method aims to achieve a compromise between accuracy and efficiency. However, when dealing with large-scale repositories, their solution encountered scaling challenges, despite the fact that it displayed enhanced performance in situations that required both semantic relevance and reliability of service concerns.

The author, who used multi-objective optimization strategies in order to solve the issue of service selection, made an additional significant contribution. Furthermore, their approach used evolutionary algorithms to concurrently maximize numerous QoS variables, so providing a complete framework for the examination of trade-offs. However, the method's convergence was dependent on the careful design of fitness functions and evolutionary

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

operators, despite the fact that it was successful in addressing competing aims effectively. On the other hand, the investigator developed a method that is based on blockchain technology in order to improve the trustworthiness and transparency of QoS ratings. With blockchain technology, their approach guaranteed the authenticity of QoS data, so addressing issues over dishonest service providers. Nevertheless, the implementation of blockchain resulted in the introduction of delay and resource cost, which restricted its application in contexts that were heavily dependent on time.

Recent studies have also focused on user-centric approaches to service selection. The investigator displayed a crowdsourcing-based approach that aimed to aggregate user evaluations and comments for better quality of service evaluations. This method improved service selection based on reliability by using real-world interactions with clients. Conversely, rigorous techniques were required to filter out untrustworthy inputs since data provided by others included possible skews and inconsistency. The author also introduced a system of suggestions for internet services. Taking into consideration users' interests and past conversations, the above system would propose services using cooperative filtering techniques. Although these systems offered personalized service choices, they struggled to handle cold-start scenarios with little user data available.

Multiple approaches are presented in the research on selecting online services, including optimized algorithms, utility-based designs, neural networks, and semantics methods, among many others. Overall, it demonstrates the variety of approaches that may be used. Because of the issue's intricacy and the different requirements of various usage areas, each technique has its own advantages and disadvantages. Conventional methods, including linear programming and evolutionary algorithms, worked wonders in controlled environments. In contrast, modern tech like blockchain technology and deep learning has expanded provider choices to include decentralised and ever-changing environments. Despite the significant advancements, several concerns regarding scalability, processing speed, and data integrity remain unresolved. Research in this field must continue because of this. Combining the best features of several approaches may be the focus of future studies in the form of hybrid designs. Technologies that are robust and adaptable to evolving service environments would be ensured by approach.

3. Objective of research work

A dependable and efficient method for selecting the top online services based on QoS criteria, including both non-functional and functional demands, is the primary objective of this research. Traditional techniques for selecting products have been cantered on matched efficiency, ignoring other important aspects of performance such as responsiveness, accessibility, reliability, and cost. The goal of this research is to lay out a comprehensive framework for ranking and evaluating online services based on a number of QoS criteria. Verifying that the selected services are as good as required in terms of functioning and efficacy is the main objective. In data-driven, fast-paced fields like marketing and finance, wherever customer satisfaction is paramount, this is of the utmost importance.

4. Motivation of research work

The requirement to select the most appropriate web services to meet real world and conceptual requirements, as well as their growing significance in modern software platforms, motivate the research in this area. As a growing number of businesses use service-oriented architectures (SOA), the challenge of identifying and selecting the most suitable internet service from an ever-expanding pool becomes even more acute. Service quality is crucial in sectors such as marketing and financial services, where data analysis, customer engagement, and financial transactions play a pivotal role. Quality of service (QoS) measures must be part of the selection process since conventional methods rely only on operational capacity and ignore issues with efficiency and reliability in real-world applications.

5. The proposed Method

Web-based service selection plays a crucial part in the deliberate alteration of services to increase customer acceptance or favourability. When a requester specifies a functional need, a number of services are found and evaluated; the process of selecting one from this pool is known as web-based service selection. In order to optimize the customers' stated utility and ensure that users are happy with the quality of services, companies are chosen to perform a company function. The delay, efficiency, availability, pricing, accessibility, integrity, effectiveness, correctness, capacity, time to execution, credibility, transmitting cost, and dependability are some of the QoS metrics that have been established. While service discovery is an essential first step in the service selection process, the real meat and potatoes of getting the right service for a requester lie in service selection. When deciding on the best service for an applicant, two primary categories of criteria include functioning and non-functional, with a focus on QoS. Effective approaches and processes are necessary for proper website service selection, the primary focus of service orientated computation, as web search for services is insufficient on its

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

own to choose a service which would meet the needs of users. Within the services the database, a collection of similar services has been chosen. Each of the service provider' WSDL files are stored in the service's repositories. In WSDL, the service name identifies the specific service, and the operation identifies the consumer-accessible technique, both of which may be utilized when processing the matching service. Just learning about the service could lead to choosing the wrong one. As a result, the activities match with the requirements, and the set of relevant services is picked, in order making the choosing procedure a lot simpler.

It is ideal to optimally fulfil the requests of many services for comparable services. Techniques such as the 0/1Knapsack issue Prior work offered genetic algorithms, memetic algorithms, and particle swarm optimization. For a given number of 57 candidates, the 0/1 Knapsack issue necessitates selection. Of some contexts, the genetic and memetic techniques of are providing subpar performance. It might seem that particle swarm optimization is a complicated and very rigorous method that seeks the optimal choice. Meeting the expectations of customers, clients, and consumers is what quality implies from an IT standpoint. A service's overall efficiency is what is known as its QoS. The specific web-based transaction's many criteria are used to determine the overall efficacy. B2B, B2C, B2G, and G2C service quality is becoming more important as businesses use web services technologies. The online services are limited by ineffective quality criteria when utilizing the web, network capacity, worldwide registries, and servers.

If a user wants to pay more to finish using the service earlier other users, WSMQ may accommodate that by giving that user's service-related message greater priority. The execution frequency of messages increases as their priority increases. Every web service is given a priority when you register. In this system, the queue server sorts' incoming request for services messages in order of priority relative to the Web service they are requesting. Messages with a higher priority are processed more often. Neither of the two prioritizing methods can ensure that all service requests will be addressed in a timely manner. Web service outages, excessive demand for particular amenities, etc., are only a few examples of the many causes of this that are above of WSMQ's control.

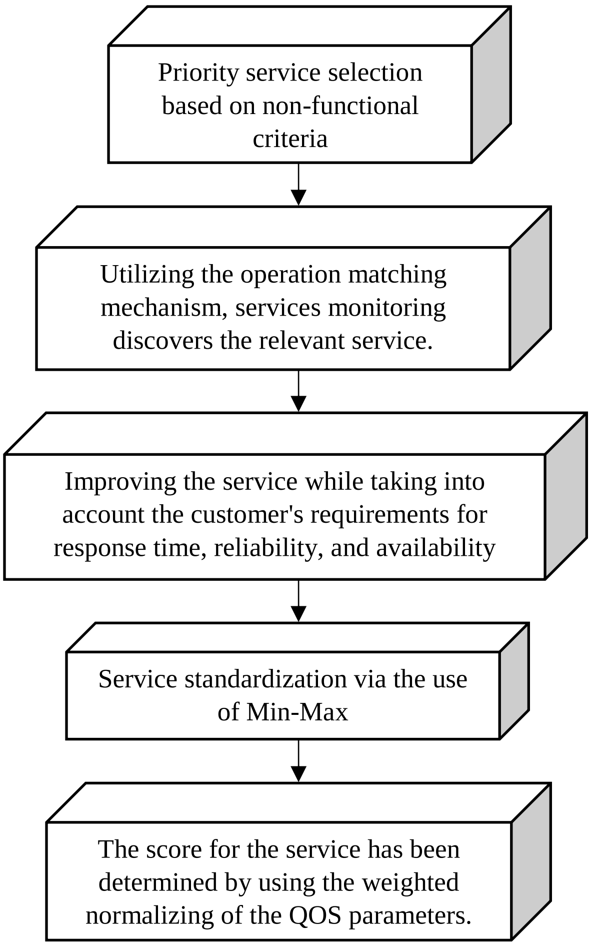

Figure 2. Flowchart of the proposed service selection procedure

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

On the other hand, the prioritizing process considerably raises the odds of quickly finishing messages of high priority. The inevitable consequence of this is that lower-priority communications will take longest to be processed. Assuming we can keep from being hungry, this is usually not a problem. Typically, there are four different kinds of quality metrics at your disposal. QoS pertaining to security, managing configuration costs, transactions, and runtime are the four categories. When it comes to service design, run-time related QoS is usually king. Efficiency, Accessibility, and Consistency are some of the factors that may be used to calculate the effectiveness.

It is possible to divide web services into three groups based on where they are physically located. What makes them Accessible using the local machine's services. The Private Company Registry offers the following services. Xmethods, Amazon, UDDI, and other similar services are accessible via library branches. Priority services selection based on non-functional criteria. Finding the relevant service via operations matching is what service tracking is all about. Service improvement based on customer-specified requirements for uptime, consistency, and speed of response The score for the product or service was calculated and determined by using a weighed normalizing of the QOS parameters. Services being standardised Using Min-Max Technique. Within a website reference, we may include the http://localhost references. For the purpose for the relevant internet techniques to be invoked and carried out. However, regardless of it is a private registration, the client may be aware of the vendor's Uniform Resource Identifier (URI), such as http://ghayathrij/webservice.example, and it will be contained in a website reference's URL. For this specific URI, it provides a list of all accessible online services. It is up to the user to choose what they want from the ones provided. A company's company procedure is the primary user of this online service type. This is a completely new situation. Reason being, the client may get an endless supply of services relevant to the service they need by using the public register. There is a plethora of more services that are also included in the register. Products of this sort may be accessed globally. More and more customers and vendors of services are using these registries and services. Customers will need to put in more work on their end of applying to locate the correct service when using this form of registration.

$$R_n = \{r | r \in S, match(E_s)\} \quad (1)$$

The service repositories are denoted by S, the set of matching services is R_n , and functional requirements are denoted by E_s .

$$P_n = \{p | p \in R_n, match(Pq_s) \quad (2)$$

With P_n standing for the set of services that have matching operations and Pq_s standing for the needed operations.

$$QoS_P = \frac{p - p_n}{p_m - p_n} \quad (3)$$

The expressions p, p_n , p_m represent the lowest and highest values of the QoS variable, respectively, where dp represents the parameter value.

$$QoS_M = \frac{p_m - p}{p_m - p_n} \quad (4)$$

$$ZQoS_M = z.QoS \quad (5)$$

Where z is a user-defined weight, QoS is the normative quality of service value, and ZQoS is the calculated normal value.

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

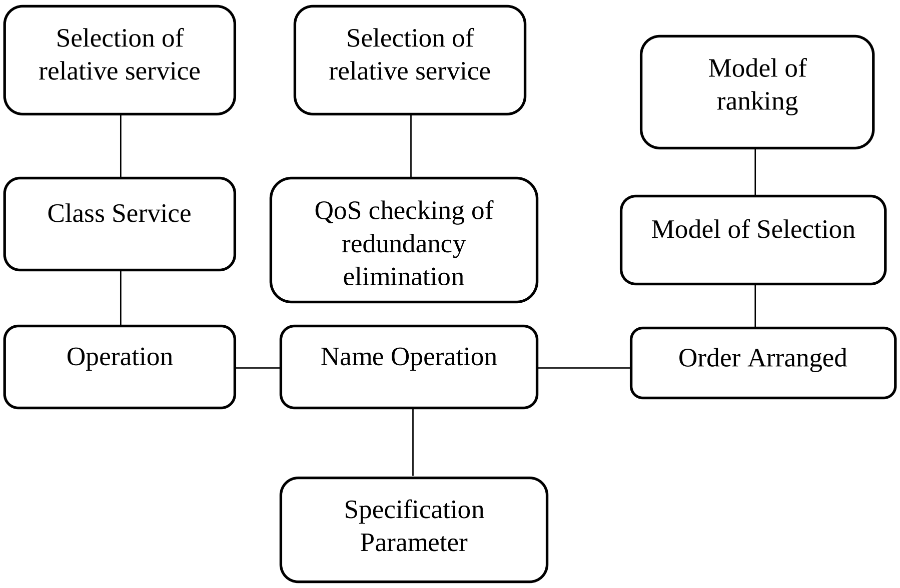

Figure 3. Service selection procedure operational flow

The term "web service selection" describes the steps used to choose one-service implementations from among several that were found in reply to a functional need provided by the applicant. In order to maximize the stated utility and satisfy the users' high-quality service requirements, services are chosen to carry out a business operation. The latency, efficiency, speed, dependability, expense, accessibility, truthfulness, effectiveness, regularity, reliability, capability, time to execution, credibility transmitting cost, and many more QoS factors have been established. The process for choosing the services is outlined in Algorithm-1. The best service is chosen by adding up all the scores, and the requirements of the customers are taken into account using the min-max boundaries of the non-functional parameters. Because, while the other three quality of service factors ought to possess maximum values, the reaction time would have the lowest value in order to pick the most excellent service.

I/P: Data Set, Limitations on Parameters

O/P: Service with top rating

Making a choice about the service

Step 1: Invoke Service Tracking (Repository, R)

Step 2: The consumer's constraints values for the QoS parameters.

Step 3: Invoke Service Refining (R, S)

Step 4: Using Equations 1 and 2, standardize the outcomes of the QoS indicators of each the customer's improved services.

Step 5: Use Equations 3 and 4 to normalize the values of the constraints of the QoS parameters.
Step 6: Determine each customer's weighted normalized numbers

Step 7: dial R for Service Grading.

Services discovery is the first step in the service choice procedure, but really, the most important thing is to solve the service selection problem so that the requester may get the right service. Optimal service selection for a requester takes into account both functional and non-functional attributes, with a focus on QoS. The primary focus of service-orientated computation is efficient methodology and methods for proper digital service choices, as web service discovering simply is insufficient for choosing favourable applications that would meet the needs of users. We have chosen a collection of similar services from our service repository. Web Services Description Language (WSDL) files are stored in the service repositories. The WSDL service identity and operations parameters allow a compatible service to be handled, with the service name identifying the specific service and

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

the action parameter describing the consumer-accessible method. Finding the service alone might lead to selecting a service that is not appropriate. Therefore, in order to streamline the choosing manage, the activities are compared to the requirements and the set of related services is chosen accordingly. For example, while looking for a service like TravelService, the first thing we do is searching for the service description and operation. The following is the format of a TravelService WSDL file.

Applying QoS characteristics to web services helps to narrow down the chosen collection of relevant services. This system takes reaction time, accessibility, efficiency, and dependability into account as QoS factors. Response One measure of a web service's efficiency is the amount of time it takes to process a request for that service. In milliseconds, it is evaluated. As a measure of the service's capacity to be maintained and its quality, durability is an important attribute of a web service. The % is the unit of measurement. In an available system, there is nobody waiting for the service. It indicates the likelihood that the solution is available and ready to be used right now. The percentage is the unit of measurement. A web-based service's effectiveness is defined as its capacity to fulfil the client's requests. A percentage is used to measure it. Users may choose certain critical criteria for QoS settings in order to enhance the system. A collection of refined services is formed by services that pass though refinements restrictions.

$$D(r) = \begin{cases} 1, & if\ p_j \geq p_n\ for\ all\ j \\ 0, & otherwise \end{cases} \quad (6)$$

Where D(r) denotes the service s constraints check while pj denotes the QoS value for the j-th variable.

$$S_r = \{r | D(r) = 1\} \quad (7)$$

$$R_j = \sum_{i=1}^{m} ZQoS_{m,n} \quad (8)$$

The overall score for service j is denoted as R_j , and the weighted quality of service value for the j-th element is denoted as $ZQoS_{m,n}$.

$$R_{optimal} = argmax\ R_m, \quad j \in S_r \quad (9)$$

$$V(r) = \sum_{j=1}^{m} z_j . v_j(r) \quad (10)$$

In this context, V(r) refers to the service's utility score, z_j denotes its weight, and $v_j(r)$ denotes the average value of the j i-th quality of services variable.

$$v_j(r) = \frac{v_j(r) - v_j^n}{v_j^n - v_j^m} \quad (11)$$

This is where the measured value of the j-th parameter is represented by $v_j(r)$.

$$V_j = \{r | p_j(r) \geq v_j\} \quad (12)$$

$$E_d = \cap_{j=1}^{m} V_j \quad (13)$$

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

When all constraints have been satisfied, the last set of services is denoted as E_d .

$$B(r) = \sum_{j=1}^{m} z_j . p_j(r) \quad (14)$$

The total weighted score for all services s is calculated using this equation. Here, the weight given to the j-th quality of service metric (such as response time or dependability) is denoted by z_j , representing the relevance of that value to the user. $p_j(r)$ is the service r-specific quantity of the j-the quality of service variable.

$$S(r) = rank\big(B(r)\big),\ r \in E_d \quad (15)$$

The equation uses the aggregate weighted score $B(r)$ to determine the rank $S(r)$ of each service. Generally speaking, services with greater $B(r)$ values are given better rankings; for example, the best rank is 1. From the set E_d , which includes all services that meet the QoS restrictions, the ranking enables users simply discover the best suited service.

$$V_s = \frac{T_E - T_S}{M_S} \quad (16)$$

In this case, the average service response time is determined by V_s . The last moments beginning and ending times M_s is the total amount of requests, while T_E and T_S are the timestamps for each transaction. A faster and better service is indicated by a lower V_s . Apps such as marketing and finance rely heavily on this statistic since delays may affect decision-making and customer satisfaction.

$$B_u = \frac{Time\ VQ}{Overall\ Time} \quad (17)$$

$$S_c = 1 - \frac{Requests\ of\ Failed}{Requests\ of\ total} \quad (18)$$

The percentage of time a service is functioning is measured by availability (B_u). The $Overall\ Time$ is the sum of all observational periods, whereas $Time\ VQ$ is the amount of time the service is available. The constancy of a service may be measured by its reliability (R b). You may get it by taking the failure rate ($Requests\ of\ Failed$, $Requests\ of\ total$) and dividing it by 1. Services that process sensitive or frequent transactions must be dependable in order to reduce the number of rejected requests and guarantee uninterrupted functioning.

6. Results Analysis

A refinement technique that makes use of sentient QoS parameters is used to choose the best online services in relation to the limitations imposed by the users. In order to increase effectiveness and customer happiness, this study takes into account both functional and non-functional criteria while choosing services.

6.1 Response Time: Indicates the expected turnaround time for a service request.

6.2 Reliability: The ability to keep the service running smoothly and well.

6.3 Availability: Service interruption during maintenance periods.

6.4 Performance: Web service's capacity to fulfil customer demands

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

Table 2: Priorities of QoS Criteria for comparing response time services

Relative Services	
Services	Response Time
Service 1	832.36
Service 2	178
Service 3	122.67
Service 4	1784.25
Service 5	220
Service 6	150.33
Service 7	250.5
Service 8	408

Response times, which show how quickly one service handles a request, are used to compare the comparable services. With a response time of only 122.67 milliseconds, Service 3 is the most effective option for applications that need a quick response. It follows most closely by Service 6 (150.33 ms) and Service 2 (178 ms), all of which have low latency and are appropriate for situations that need fast answers. In contrast, the response times of Services 1, 4, and 8 are much greater at 832.36 ms, 1784.25 ms, and 408 ms, respectively, which may affect their usefulness in applications that run in real time. Two services in the middle of the pack, 5 (220 ms) and 7, (250.5 ms), provide satisfactory results. When it comes to low latency needs, Service 3 is clearly the best option. Although Services 2 and 6 are also good choices, the other services could be better suited for jobs that aren't as time-sensitive.

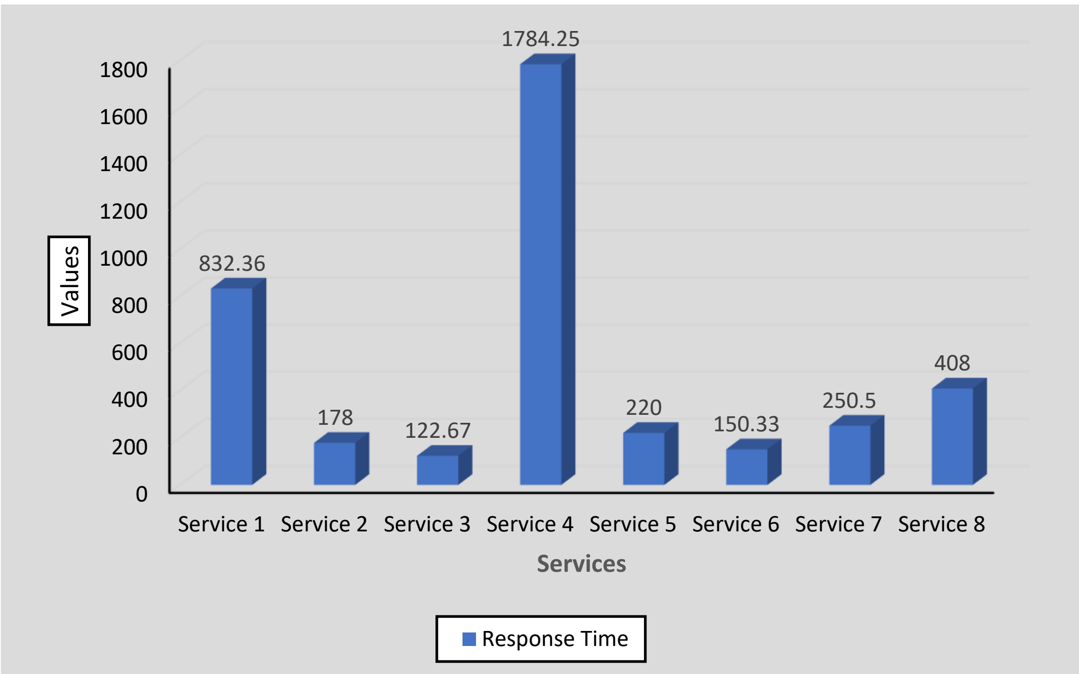

Figure 4. Assessment of ML models in relation to conventional methods.

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

Table 3: Priorities for Comparing Service Performance Dependent on QOS Requirements

Relative Services	
Services	Availability
Service 1	65
Service 2	89
Service 3	100
Service 4	94
Service 5	88
Service 6	90
Service 7	96
Service 8	84

Comparisons are made between the services according to their availability, a key quality of service metric that represents the probability of a service existing functional. Service 3, with an availability score of 100%, is the most dependable option for applications that need continuous service since it is constantly available. Both Service 4 (94%) and Service 7 (96%) are highly available, guaranteeing little downtime and strong performance. The somewhat reduced but consistent availability offered by services 2 (89%) and 6 (90%) is ideal for situations where the occasional downtime is tolerable. Service 5's availability is modest at 88% and Service 8's availability is moderate at 84%, which may make them unsuitable for uses requiring high dependability. Service 1 is not recommended for applications that are critical due to its poor availability of 65%. When it comes to availability, Service 3 is far above the competition. Services 7, 4, and 6 are all formidable candidates for use cases that prioritize dependability.

Figure 5. Evaluate ML models in comparison to more traditional approaches.

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

Table 4: Performance-Related of QoS Parameter Values

Relative Services	
Services	Performance
Service 1	66
Service 2	97
Service 3	100
Service 4	94
Service 5	97
Service 6	90
Service 7	99
Service 8	85

Service 3 is the best option for activities that need optimum execution since it shows the maximum performance at 100%. After it comes Service 7 with 99% and Services 2 and 5 with 97%, all of which provide top-notch performance that is ideal for critical tasks. Services 4 and 6 provide considerably less nevertheless reliable efficiency at 94% and 90%, respectively, whilst Services 1 and 8 are not adequate when it comes to outstanding durability requirements, falling behind at 66% and 85%. In terms of overall performance, Service 3 is clearly the best, with Services 7, 2, and 5 all providing solid possibilities.

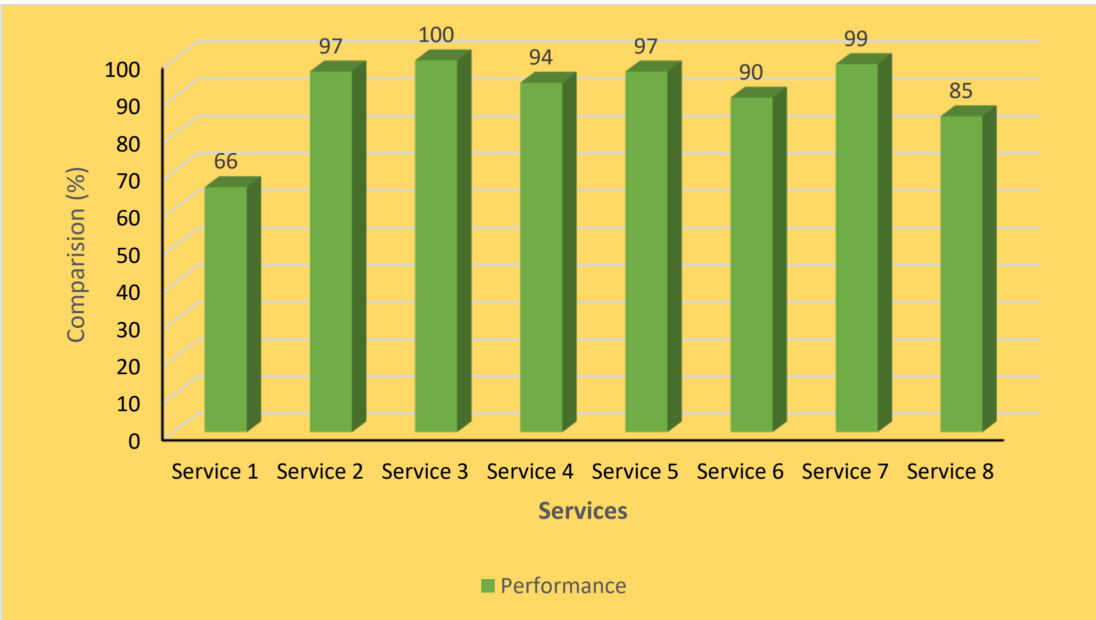

Figure 6. Efficiency of different methods

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

Table 5: Investigative DL models in connection to suggested approaches

Relative Services	
Services	Reliability
Service 1	70
Service 2	74
Service 3	75
Service 4	56
Service 5	74
Service 6	74
Service 7	74
Service 8	68

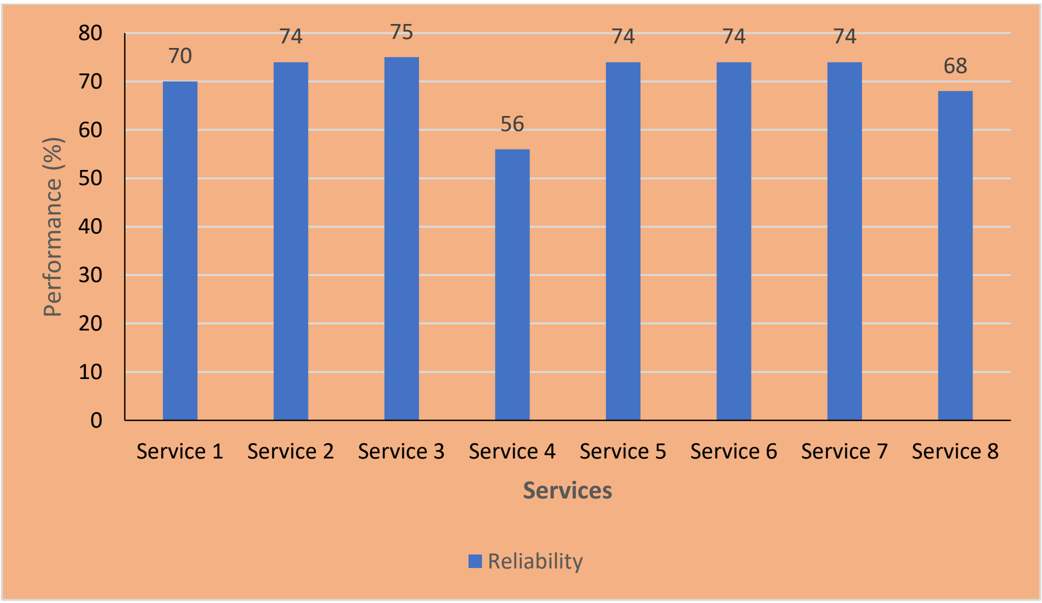

Figure 6. Effectiveness of different models

When it comes to constant functioning, Service 3 is your best bet with a dependability rate of 75%. Offering good dependability for jobs that demand stability, Services 2, 5, 6, and 7 follow closely after at 74% each. When compared to the other services, Service 1's (70%) dependability is significantly below average. With a reliability score of just 68%, Service 8 is clearly not a reliable choice for mission-critical tasks. Even though Services 2, 5, 6, and 7 are all viable choices, the most dependable one is Service 3.

7. Conclusion

Taking into consideration both functional and aesthetically requirements, this part sets forth a sound methodology for selecting the top online services according to QoS attributes. In order to guarantee that the selected services meet user requests by providing the necessary performance indicators, this method considers critical quality of service elements such as response time, availability, and reliability. This method really comes

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

into its own in fast-paced fields like marketing and banking, where providing high-quality service in real-time is crucial. The proposed framework offers a reliable way for selecting and assessing services based on user-defined criteria, and it incorporates service discovery, items refining, and grading. Businesses have greater freedom to tailor the service selection process to their own needs when they utilize algorithms to analyse and standardize QoS criteria. Investigations prove that the strategy delivers the expected results, recommending reliable services to suit various customer types. In the future, researchers may employ machine learning and artificial intelligence for automation and make the product selection process more flexible. These innovations provide real-time adjustments to service effectiveness, which can be useful in dynamic industries like marketing and financial services. Integrating blockchain technology for trustworthy, secure high-quality service evaluations and enhancing scalability to handle massive service repositories can boost economy and trust.

To further enhance the technique for specialized sectors, it is worth exploring the incorporation of domain-specific QoS factors. Bridging the gap among studies and practical deployments by creating user-friendly interfaces for smooth execution of these approaches in real-world scenarios. This would make the procedure accessible to corporations. For optimum judgments, decision-makers ought to take seriously various unique elements; future research might apply multi-criteria decision-making procedures to assess new product marketing tactics.

Funding: "This research received no external funding"

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

References

[1] M. Hosseinzadeh, H. K. Hama, M. Y. Ghafour, M. Masdari, O. H. Ahmed, and H. Khezri, "Service selection using multi-criteria decision making: A comprehensive overview," J. Netw. Syst. Manag., vol. 28, pp. 1639–1693, Jul. 2020.

[2] L. Huang and S. Deng, "Service selection for mobile service orchestration," in Proc. IEEE Int. Conf. Mobile Services, 2014, pp. 147–148.

[3] S. Zaman et al., "Security threats and artificial intelligence-based countermeasures for Internet of Things networks: A comprehensive survey," IEEE Access, vol. 9, pp. 94668–94690, 2021.

[4] H. K. Apat, R. Nayak, and B. Sahoo, "A comprehensive review on Internet of Things application placement in fog computing environment," Internet Things, vol. 23, Oct. 2023, Art. no. 100866.

[5] M. K. Alhassan, A. A. Al-Fuqaha, and A. A. Badawi, "Blockchain-enabled security architecture for privacy-preserving IoT applications," IEEE Internet of Things Journal, vol. 8, no. 3, pp. 2135–2145, 2021.

[6] A. S. Salama, E. A. Ghoneim, and M. A. Ragab, "A comprehensive review of IoT-based security and privacy techniques," Journal of Cybersecurity and Privacy, vol. 2, no. 1, pp. 49–70, 2021.

[7] C. Muralidharan and R. Anitha, "EDSAC–an efficient Dempster Shafer algorithm for classification to estimate the service, security and privacy risks with the service providers," Wireless Pers. Commun., vol. 122, pp. 3649–3669, Feb. 2022.

[8] A. P. Mdee, M. T. R. Khan, J. Seo, and D. Kim, "Security compliant and cooperative pseudonyms swapping for location privacy preservation in VANETs," IEEE Trans. Veh. Technol., vol. 72, no. 8, pp. 10710–10723, Aug. 2023.

[9] J. S. R. Prasanna, V. V. V. S. L. Prasad, and S. Chandra, "Service selection and orchestration in edge computing for IoT applications," Journal of Cloud Computing: Advances, Systems and Applications, vol. 11, pp. 24–41, 2021.

[10] D. K. S. Gupta, A. S. Kumari, and D. L. Rathi, "Blockchain-based authentication model for securing web services," Journal of Network and Computer Applications, vol. 174, pp. 102946, 2021.

[11] M. G. A. Malik, S. K. Ray, Z. Bashir, and A. Mughal, "Selecting ubiquitous services in future heterogeneous wireless networks using multi-attributes decision making," in Proc. 11th Int. Conf. Mobile Comput. Ubiquitous Netw. (ICMU), 2018, pp. 1–4.

[12] Shuping Ran, "A Framework for discovering web services with desired Quality of Services attributes," IEEE International Conference on Web Services, Las Vegas, Nevada, USA, June 2003.

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025

[13] R. B. Patel, S. S. Patil, and S. S. Bansal, "Secure routing in wireless sensor networks using fuzzy logic and cryptography," *Comput. Commun.*, vol. 167, pp. 118–126, Jan. 2021.

[14] P. L. B. Kumar, D. A. K. Parab, and T. N. Zeng, "Fuzzy logic based secure routing in wireless sensor networks," Wireless Communications and Mobile Computing, vol. 2021, Article ID 9865264, 2021.

[15] V. Tosic, B. Pagurek, K. Patel, "WSOL: A language for the formal specification of classes of service for web services," International Conference on Web Services, Las Vegas, Nevada, USA, June 2003.

[16] Hongan Chen, Tao Yu, Kwei-Jay Lin, "QCWS: An implementation of QoS-capable multimedia web services," IEEE Fifth International Symposium on Multimedia Software Engineering, December 2003.

[17] F. S. Alharbi, A. H. Alqahtani, and M. A. Alzahrani, "AI-based security for IoT in smart healthcare systems: A survey and future directions," Journal of Ambient Intelligence and Humanized Computing, vol. 13, no. 8, pp. 3661–3679, 2022.

[18] A. Ahmed, M. A. Alam, and R. R. J. L. Sant, "An ensemble learning approach for facial emotion recognition using convolutional neural networks," Computers, Materials & Continua, vol. 68, no. 2, pp. 1799–1813, 2021.

[19] W. T. Tsai, R. Paul, Z. Cao, L. Yu, A. Saimi, B. Xiao, "Verification of web services using an enhanced UDDI server," Eighth IEEE International Workshop on Object-Oriented Real Time Dependable Systems, Guadalajara, Mexico, January 2003.

[20] K. Khadir, N. Guermouche, T. Monteil, and A. Guittoum, "Towards avatar-based discovery for IoT services using social networking and clustering mechanisms," in Proc. 16th Int. Conf. Netw. Service Manage. (CNSM), Nov. 2020, pp. 1–7.

[21] Y. Liu, "Service selection method based on skyline in cloud environment," Int. J. Performability Eng., pp. 1039–1047, 2017.

[22] M. Rajeswari, G. Sambasivam, N. Balaji, M. S. S. Basha, T. Vengattaraman, and P. Dhavachelvan, "Appraisal and analysis on various web service composition approaches based on QoS factors," J. King Saud Univ.-Comput. Inf. Sci., vol. 26, no. 1, pp. 143–152, Jan. 2014.

[23] T. Yu, Y. Zhang, and K.-J. Lin, "Efficient algorithms for web services selection with end-to-end QoS constraints," ACM Trans. Web, vol. 1, no. 1, p. 6, May 2007.

[24] V. R. Chifu, C. B. Pop, I. Salomie, and E. S. Chifu, "Hybrid honey bees mating optimization algorithm for identifying the near-optimal solution in web service composition," Comput. Informat., vol. 36, no. 5, pp. 1143–1172, 2017.

[25] G. Chiandussi, M. Codegone, S. Ferrero, and F. E. Varesio, "Comparison of multi-objective optimization methodologies for engineering applications," Comput. Math. Appl., vol. 63, no. 5, pp. 912–942, 2012. [Online]. Available: http://www.sciencedirect.com/ science/article/ pii/ S0898 122 1 11 010406.

Doi: https://doi.org/10.54216/IJAACI.070103
Received: September 27, 2024 Revised: December 02, 2024 Accepted: January 11, 2025